

Ατομική Διπλωματική Εργασία

***TelcoWare*: ΜΙΑ ΕΠΕΚΤΑΣΙΜΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΓΙΑ ΤΗΝ
ΕΠΕΞΕΡΓΑΣΙΑ ΜΕΓΑΛΩΝ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΩΝ
ΔΕΔΟΜΕΝΩΝ**

Αναστασίου Χρυσοβαλάντης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2015

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

***TelcoWare*: ΜΙΑ ΕΠΕΚΤΑΣΙΜΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΓΙΑ ΤΗΝ ΕΠΕΞΕΡΓΑΣΙΑ ΜΕΓΑΛΩΝ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΔΕΔΟΜΕΝΩΝ**

Αναστασίου Χρυσοβαλάντης

Επιβλέπων Καθηγητής

Δημήτρης Ζεϊναλιπούρ

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των
απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του
Πανεπιστημίου Κύπρου

Μάιος 2015

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω θερμά τον κ. Ζεϊναλιπούρ Δημήτρη επιβλέπων καθηγητή της παρούσας διπλωματικής εργασίας για την υποστήριξη, καθοδήγηση και συμβουλές που μου παρείχε καθ' όλη τη διάρκεια της εκπόνησης της διπλωματικής μου εργασίας. Επίσης θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτορα Πληροφορικής και φίλο Κωνσταντίνο Κώστα για όλη την βοήθεια και συνεχή καθοδήγηση που μου προσέφερε σε διάφορα τεχνικά θέματα.

Ακολούθως θα ήθελα να ευχαριστήσω τους φίλους μου, εντός και εκτός Πανεπιστημίου, που μου συμπαραστάθηκαν όλο αυτό τον καιρό αλλά και ένα τεράστιο ευχαριστώ στην οικογένεια μου που συνεχώς με ενθάρρυνε να συνεχίσω και να μην πτοούμαι από δυσκολίες και εμπόδια που έβρισκα μπροστά μου. Ειδικότερα, θα ήθελα να ευχαριστήσω τους γονείς μου για την ηθική και οικονομική υποστήριξη που μου προσέφεραν καθ' όλη την διάρκεια των σπουδών μου.

Τέλος, θα ήθελα να ευχαριστήσω τις φίλες και συμφοιτήτριες μου Ζιανέτ Σιαχουάν και Στέλλα Κωνσταντίνου για τις στιγμές χαλάρωσης που μου χάρισαν μεταξύ των αμέτρητων ωρών μελέτης που αφιέρωνα στην έρευνα για την διπλωματική μου εργασία.

Αναγνώριση

Η πιο κάτω δημοσίευση προέκυψε από τεχνογνωσία που αποκτήθηκε μέσω αυτής της διπλωματικής εργασίας:

"Rayzit: An Anonymous Dynamic Social Network Architecture", C. Costa, **C. Anastasiou**, G. Chatzimilioudis, D. Zeinalipour-Yazti "3rd International Workshop on Mobile Data Management, Mining, and Computing on Social Networks" (MobiSocial'15), Pittsburgh, Pennsylvania, USA, June 15, 2015

Περίληψη

Στα πλαίσια αυτής της διπλωματικής εργασίας παρουσιάζεται το TelcoWare, μια επεκτάσιμη αρχιτεκτονική για την επεξεργασία μεγάλων τηλεπικοινωνιακών δεδομένων. Τα δεδομένα αυτά παράγονται από διάφορα συστήματα που διαθέτουν οι παροχείς μέσα από τα οποία περνούν όλα τα δεδομένα από τις ενέργειες των χρηστών του δικτύου (π.χ., τηλεφωνικές κλήσεις, γραπτά μηνύματα, συνεδρίες διαδικτύου κ.α.).

Τα δεδομένα αυτά αποθηκεύονται με σκοπό κυρίως την αξιοποίησή τους για υπολογισμό των χρεώσεων των χρηστών. Τον τελευταίο καιρό όμως φαίνεται να υπάρχει μια ανάγκη για περαιτέρω ανάλυση αυτών των δεδομένων και αξιοποίησης της πληροφορίας που έμμεσα μεταφέρουν προς βελτιστοποίηση των κερδών του τηλεπικοινωνιακού οργανισμού.

Όπως εύκολα μπορεί να αντιληφθεί κανείς, πολύ γρήγορα αυτά τα δεδομένα που συλλέγει ένας παροχέας αποκτούν μεγάλο όγκο. Τα παραδοσιακά εργαλεία διαχείρισης δεδομένων δεν είναι πλέον ικανά να διαχειριστούν τον μεγάλο όγκο δεδομένων σε επιτρεπτά χρονικά περιθώρια.

Το TelcoWare εκμεταλλεύεται τις δυνατότητες του κατανεμημένου εργαλείου Apache Spark και τις συνδυάζει με μερικές αφαιρετικές κλάσεις μέσω των οποίων προσφέρει μια απλή προγραμματιστική διεπαφή για την εύκολη ανάπτυξη καινούργιων εργασιών ανάλυσης οι οποίες εκτελούνται κατανεμημένα σε ένα συγκρότημα υπολογιστικών μονάδων.

Στα πλαίσια αυτής της διπλωματικής εργασίας υλοποιήθηκαν μερικοί τελεστές υψηλού επιπέδου και δοκιμάστηκαν με συνθετικά δεδομένα για την απόδοσή τους ούτως ώστε να αποτιμηθεί η απόδοση που προσφέρει το σύστημα.

Τα αποτελέσματα μπορούν να ανακτώνται και να παρουσιάζονται με γραφικό τρόπο έτσι ώστε να γίνονται πιο εύκολα κατανοητά από το διοικητικό και τεχνικό προσωπικό του οργανισμού. Η αξιοποίηση των αποτελεσμάτων συμβάλει στη λήψη αποφάσεων στον οργανισμό οι οποίες αποσκοπούν στη βελτιστοποίηση των κερδών.

Τέλος, αξίζει να σημειωθεί πως αν και βρίσκεται σε πρώιμο στάδιο, εντούτοις, το σύστημα TelcoWare, είναι μια επεκτάσιμη αρχιτεκτονική με την έννοια του ότι επιτρέπει την αυθαίρετη πρόσθεση εργασιών ανάλυσης και παρουσίασης των αποτελεσμάτων.

Περιεχόμενα

Ευχαριστίες.....	iii
Αναγνώριση	iv
Περίληψη	v
Περιεχόμενα.....	vi
Κεφάλαιο 1 Εισαγωγή	1
1.1 Υποκίνηση εργασίας	1
1.2 Ανασκόπηση Συστήματος TelcoWare	4
1.3 Συνεισφορές.....	5
1.4 Περίγραμμα Εργασίας	6
Κεφάλαιο 2 Γνωσιολογικό Υπόβαθρο.....	7
2.1 Αρχιτεκτονική και Συστήματα Τηλεπικοινωνιακών Παροχών	7
2.1.1 Κεραίες Κυψέλης.....	8
2.2 Δεδομένα Τηλεπικοινωνιακών Παροχών.....	9
2.2.1 Call Detail Record (CDR).....	9
2.2.2 Network Measurement Report (NMR)	10
Κεφάλαιο 3 Σχετική Βιβλιογραφία	11
3.1 Πρόβλεψη των Χρηστών που θα Εγκαταλείψουν την Υπηρεσία.....	11
3.1.1 Η Αρχιτεκτονική Συστημάτων Πρόβλεψης Εγκατάλειψης Υπηρεσίας	12
3.2 Huawei – Εξειδικευμένη Ανάλυση Μεγάλων Τηλεπικοινωνιακών Δεδομένων ..	13
3.2.1 Η Λύση Ανάλυσης Μεγάλων Δεδομένων της Huawei.....	14
3.3 Γεωτοποθέτηση Κινητών Συσκευών στα Τηλεπικοινωνιακά Δίκτυα GSM	15
3.4 OceanRT: Ανάλυση Μεγάλων Χρονικών Δεδομένων σε Πραγματικό Χρόνο	15
3.5 Αναλύοντας Έξι Δισεκατομμύρια CDR την Ημέρα.....	16
3.6 Data for Development (D4D)	17
3.6.1 Η Πρόκληση	18
3.6.2 Συστήματα που Προτάθηκαν και Παρουσιάστηκαν.....	18
Κεφάλαιο 4 Αρχιτεκτονική του TelcoWare	20
4.1 Τεχνικό Υπόβαθρο.....	20
4.1.1 Μεγάλα Δεδομένα και Ροές Δεδομένων (Big Data & Data Streams)	21

4.1.2 Κατανεμημένη και Παράλληλη Επεξεργασία	22
4.1.3 Εργαλεία Διαχείρισης Μεγάλων Δεδομένων.....	25
4.2 Υποδομή Ραχοκοκαλιάς.....	33
4.3 Γεννήτρια Προσομοίωσης Δεδομένων CDR.....	35
4.4 Συστατικό TelcoWare Master.....	35
4.5 Εργασίες Επεξεργασίας TelcoWare Jobs	36
Κεφάλαιο 5 Τελεστές Υψηλού Επιπέδου στο Telcowaare	38
5.1 Υπολογισμός Συνδεδεμένων Χρηστών ανά Κεραία Κυψέλης.....	39
5.2 Ευρετήριο Ιστορικού Κλήσεων για Κάθε Χρήστη	42
5.3 Χρονικό Ευρετήριο Χρηστών ανά Περιοχή	43
5.4 Χρονικό Ευρετήριο Αριθμού Χρηστών στο Δίκτυο.....	45
Κεφάλαιο 6 Εφαρμογές Προβολής και Οπτικοποίησης	47
6.1 Επίπεδο Εξυπηρετητών.....	47
6.1.1 TelcoWare API	49
6.1.2 Συστατικά Εξυπηρετητών (Server Modules).....	49
6.2 Επίπεδο Δεδομένων	50
6.3 TelcoWare Viewer	52
Κεφάλαιο 7 Πειραματική Αποτίμηση Συστήματος.....	55
7.1 Πειραματική Μεθοδολογία.....	55
7.2 Απόδοση Αναλύσεων	56
Κεφάλαιο 8 Συμπεράσματα και Μελλοντικές Επεκτάσεις	61
8.1 Συμπεράσματα	61
8.2 Μελλοντικές Επεκτάσεις	63
Βιβλιογραφία	65

Κεφάλαιο 1

Εισαγωγή

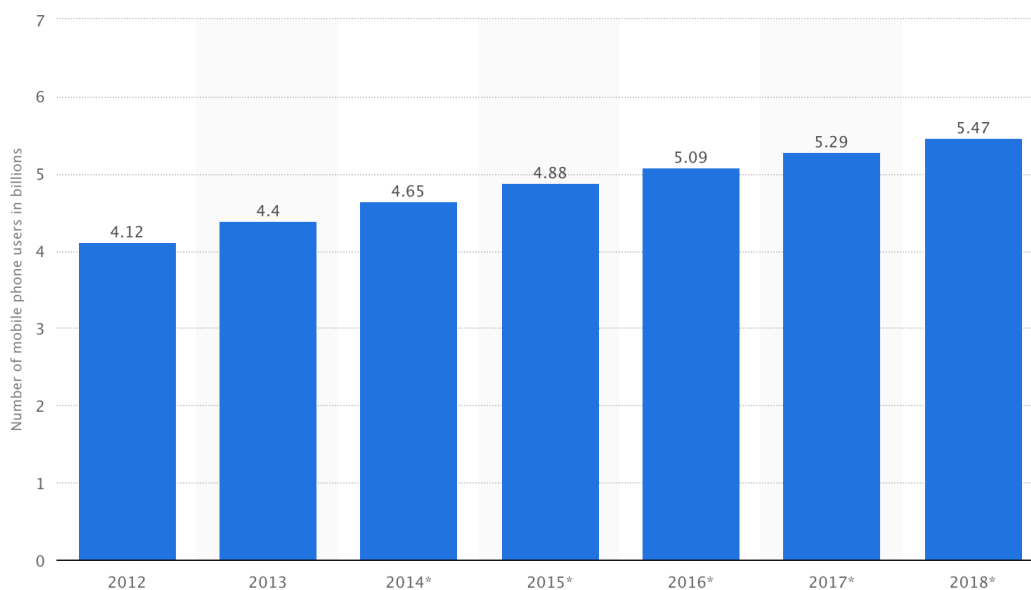
1.1	Υποκίνηση εργασίας	1
1.2	Ανασκόπηση Συστήματος TelcoWare	4
1.3	Συνεισφορές	5
1.4	Περίγραμμα Εργασίας	6

1.1 Υποκίνηση εργασίας

Τα τελευταία χρόνια παρατηρείται μια σημαντική αύξηση στον αριθμό των ατόμων που χρησιμοποιούν κινητές συσκευές. Σύμφωνα με μελέτη που έχει γίνει από τον οργανισμό Statista¹ και παρουσιάζεται στο *Σχήμα 1.1* μέχρι το 2018 ο αριθμός των ενεργών κινητών συσκευών υπολογίζεται να φτάσει τα 5,47 δισεκατομμύρια. Η συνεχής αυτή αύξηση αναγκάζει τους τηλεπικοινωνιακούς παροχείς να προσθέτουν συνεχώς καινούργιο εξοπλισμό ούτως ώστε να είναι δυνατή η υποστήριξη όλο και περισσότερων χρηστών. Όσο όμως προστίθεται εξοπλισμός η αρχιτεκτονική γίνεται πιο περίπλοκη. Επίσης, καθώς αυξάνεται ο αριθμός των χρηστών, αυξάνεται και ο όγκος των δεδομένων που ρέουν στο σύστημα φτάνοντας έτσι μέχρι και σε σημεία που δεν μπορούν πλέον να διαχειρίζονται από τα παραδοσιακά συστήματα των παροχέων.

Κάθε καταναλωτής ψάχνει πάντα να βρει την υπηρεσία ή το αγαθό που επιθυμεί στην καλύτερη δυνατή ποιότητα με την χαμηλότερη δυνατή τιμή. Βέβαια συχνά

¹ <http://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/>



Σχήμα 1.1 Ο αριθμός των κατόχων κινητού τηλεφώνου τα τελευταία χρόνια και πρόβλεψη για το μέλλον

παρατηρούνται συμβιβασμοί μεταξύ τιμής και ποιότητας αφού το κόστος είναι ένας από τους σημαντικότερους παράγοντες για την επιλογή του καταλληλότερου προϊόντος.

Αντίστοιχα, οι διάφοροι πωλητές ανταγωνίζονται ούτως ώστε να προσελκύσουν τους περισσότερους πελάτες μεγιστοποιώντας τα κέρδη τους. Η ίδια λογική ισχύει και στις τηλεπικοινωνίες όπου από την μια πλευρά οι πελάτες ψάχνουν και υπηρεσίες υψηλής ποιότητας, και συχνά τελευταίας τεχνολογίας, με όσο το δυνατό χαμηλότερες τιμές, ενώ από την άλλη πλευρά οι τηλεπικοινωνιακοί παροχείς στοχεύουν στην μεγιστοποίηση του κέρδους.

Στην περίπτωση των τηλεπικοινωνιακών υπηρεσιών παρατηρείται ότι οι πελάτες είναι πρόθυμοι να αυξήσουν ελάχιστα τον πήχη του κόστους όταν πρόκειται για βελτίωση της ποιότητας, ειδικά στις υπηρεσίες διαδικτύου. Επίσης, οι παροχείς είναι πρόθυμοι να προσφέρουν υπηρεσίες σε πολύ καλές τιμές για να προσελκύσουν πελάτες και να επιβιώσουν στο ανταγωνιστικό περιβάλλον τους.

Για να μην επηρεαστεί ο απώτερος σκοπός μεγιστοποίησης του κέρδους, πρέπει η προσφορά χαμηλών τιμών σε σχέση με την ποιότητα και την ποσότητα υπηρεσιών που προσφέρεται να αποφασίζεται έξυπνα. Οι παροχείς ήδη αποθηκεύουν πολλά δεδομένα σχετικά με τις ενέργειες των πελατών τους και την κατάσταση του δικτύου τους όμως

αυτά εκμεταλλεύονται μόνο κατά την έκδοση των τιμολογίων των πελατών. Η ανάλυση αυτών των δεδομένων όμως, συνήθως, παίζει σημαντικό ρόλο στη λήψη αποφάσεων αφού επιδεικνύει τις ελλείψεις στις υπηρεσίες που προσφέρονται.

Εύκολα μπορεί να αντιληφθεί κάποιος ότι πολύ σύντομα, τα δεδομένα που αποθηκεύουν οι παροχείς αποκτούν τεράστιο όγκο. Η πρώτη αναφορά σε δεδομένα τέτοιου μεγέθους έγινε μόλις πριν μερικά χρόνια με τον όρο Μεγάλα Δεδομένα (Big Data). Ο όρος αυτός αναφέρεται σε δεδομένα μεγέθους μεγαλύτερου από αυτού που μπορούν να επεξεργαστούν και να αποθηκεύσουν τα κοινά παραδοσιακά συστήματα σε αποδεχτό χρονικό πλαίσιο. Κοιτώντας πίσω στο 2001, τα Μεγάλα Δεδομένα είχαν χαρακτηριστεί από τον Doug Laney² ως τα 3-V, από τα αρχικά των Volume – Velocity – Variety. Η ιδέα που ήθελε να μεταδώσει ήταν πως τα Μεγάλα Δεδομένα έχουν ασυνήθιστα μεγάλο όγκο (Volume), παράγονται σε απίστευτα γρήγορους ρυθμούς (Velocity) και δεν έχουν κάποια συγκεκριμένη μορφή αλλά διαφέρουν από πηγή σε πηγή (Variety).

Τα τρία αυτά χαρακτηριστικά των Μεγάλων Δεδομένων καθιστούν δυσκολότερη την ανάλυση τους και απαιτούν τεχνικές πιο εξειδικευμένες για την αποδοτική τους διαχείριση. Εντούτοις, η ανάλυση τους καλύπτει τεχνικές επιχειρηματικής ευφυΐας οι οποίες ικανοποιούν περίπλοκες απαιτήσεις και συνεισφέρουν κατά την λήψη αποφάσεων εντός του τηλεπικοινωνιακού οργανισμού. Η υιοθέτηση της ανάλυσης δεδομένων μπορεί να βοηθήσει στην βελτίωση των συστημάτων και του δικτύου του παροχέα αλλά και στην πώληση των υπηρεσιών του.

Είναι σημαντικό για κάποιο παροχέα να αναπτύξει σχέσεις εμπιστοσύνης και αφοσίωσης με τους πελάτες του έτσι ώστε να αποκομίσει το μεγαλύτερο δυνατό κέρδος από αυτό. Πέραν από τις διάφορες προσφορές που μπορεί να προτείνει μαζικά, ένας οργανισμός θα μπορούσε, μέσω ανάλυσης του ιστορικού του πελάτη, να προτείνει πιο εξατομικευμένες ευκαιρίες. Αυτό αποσκοπεί στη μείωση της πιθανότητας κάποιος πελάτης να υποκύψει σε προσφορά κάποιου ανταγωνιστή.

Συνοψίζοντας, τα οφέλη που μπορούν να προκύψουν από την ανάλυση τηλεπικοινωνιακών δεδομένων έχουν τεράστια αξία. Όχι μόνο μπορούν να βοηθήσουν τον οργανισμό στη λήψη σημαντικών επιχειρησιακών αποφάσεων, αλλά επίσης μπορούν να βοηθήσουν στην κατανόηση της συμπεριφοράς των πελατών και στην πρόβλεψη

² Big Data – Wikipedia http://en.wikipedia.org/wiki/Big_data

μελλοντικών ενεργειών από ομάδες χρηστών. Η παρούσα διπλωματική στοχεύει στην ανάπτυξη ενός ολοκληρωμένου συστήματος επεξεργασίας τηλεπικοινωνιακών δεδομένων σε σχεδόν πραγματικό χρόνο μέσω της χρήσης κατάλληλων εργαλείων. Το σύστημα σχεδιάζεται να προσφέρει ένα αφαιρετικό επίπεδο καθορισμού εργασιών ανάλυσης έτσι ώστε να μπορεί εύκολα να αξιοποιηθεί ακόμα και από άτομα που δεν είναι εξοικειωμένα με την αποδοτική επεξεργασία Μεγάλων Δεδομένων.

1.2 Ανασκόπηση Συστήματος TelcoWare

Στα πλαίσια αυτής της διπλωματικής εργασίας θα παρουσιαστεί το σύστημα TelcoWare, μια επεκτάσιμη αρχιτεκτονική για την ανάλυση και οπτικοποίηση μεγάλων τηλεπικοινωνιακών δεδομένων. Επιπλέον, η διπλωματική αυτή αποσκοπεί στην υλοποίηση μερικών συναθροιστικών τελεστών ψηλού επιπέδου αλλά και χρήσιμων ευρετηρίων.

Πιο συγκεκριμένα, έχει μελετηθεί, σχεδιαστεί και υλοποιηθεί μια υποδομή κατάλληλη για την διαχείριση δεδομένων μεγάλου όγκου η οποία αποτελείται από ένα συγκρότημα διασυνδεδεμένων εξυπηρετητών. Οι εξυπηρετητές είναι σε θέση να προσφέρουν τους πόρους που διαθέτουν, μέσω ενός συντονιστή διαμοιρασμού πόρων (Hadoop YARN), για την κατανεμημένη εκτέλεση διάφορων εργασιών ανάλυσης μεγάλων τηλεπικοινωνιακών δεδομένων αλλά και επίσης για μια προγραμματιστική διεπαφή εφαρμογών (Web 2.0 API) μέσω της οποίας θα ανακτώνται τα αποτελέσματα των αναλύσεων για σκοπούς παρουσίασης. Επιπλέον, οι εξυπηρετητές του συστήματος υποστηρίζουν μια κατανεμημένη NoSQL βάση δεδομένων (Couchbase) και ένα κατανεμημένο σύστημα διαχείρισης αρχείων (HDFS).

Εκτός από το διαχειριστικό μέρος του συστήματος (back-end) έχει επίσης αναπτυχθεί και η διαδικτυακή εφαρμογή TelcoWare Viewer. Η εφαρμογή αυτή απαιτεί την χρήση περιηγητή που υποστηρίζει το πρότυπο HTML5 και είναι υπεύθυνη να ανακτά μέσω του API όσα αποτελέσματα μπορούν να παρουσιαστούν σε κάποιο χάρτη και να τα οπτικοποιεί μέσω μιας φιλικής και εύχρηστης διεπαφής.

1.3 Συνεισφορές

Μέσω αυτής της ατομικής διπλωματικής εργασίας έχουν γίνει οι εξής σημαντικές συνεισφορές:

1. Πρότεινα, σχεδίασα και υλοποίησα μια ολοκληρωμένη αρχιτεκτονική για την επεξεργασία τηλεπικοινωνιακών δεδομένων σε πραγματικό χρόνο. Η προτεινόμενη αρχιτεκτονική εισάγει έννοιες γεωγραφικών συστημάτων διαχείρισης δεδομένων, καθώς επίσης έννοιες χωροχρονικών δεδομένων για να προσφέρει πλούσια σημασιολογία για τα άμορφα παραγόμενα δεδομένα.
2. Υλοποιήθηκαν συναθροιστικοί τελεστές ψηλού επιπέδου και τελεστές για κτίσιμο ευρετηρίων για την πρώτη δοκιμή του συστήματος σε περιβάλλον πραγματικού τηλεπικοινωνιακού παροχέα.
3. Υλοποιήθηκαν εφαρμογές που παρέχουν τις υπηρεσίες που προσφέρει το TelcoWare και δίνουν μια γραφική αποτίμηση στα αποτελέσματα που προκύπτουν από την ανάλυση των διάφορων δεδομένων. Συγκεκριμένα υλοποιήθηκε η εφαρμογή TelcoWare Viewer η οποία είναι προσβάσιμη από οποιοδήποτε περιηγητή υποστηρίζει την τεχνολογία HTML5.
4. Διεξήχθηκε πειραματική μελέτη για την αποτίμηση της απόδοσης και της ευκολίας χρήσης του συστήματος. Συγκεκριμένα, αποτιμήθηκε ο χρόνος εκτέλεσης για την απόδοση και οι γραμμές κώδικα για την ευκολία χρήσης του συστήματος από το τεχνικό τμήμα του παροχέα.

1.4 Περίγραμμα Εργασίας

Στο πρώτο κεφάλαιο παρουσιάσαμε συνοπτικά την υποκίνηση αυτής της διπλωματικής εργασίας και τις συνεισφορές που έχουν γίνει προς ολοκλήρωση της.

Στο δεύτερο κεφάλαιο θα γίνει επεξήγηση εννοιών και ορισμών που αφορούν τηλεπικοινωνιακά δεδομένα και χρησιμοποιούνται εκτεταμένα κατά την περιγραφή του συστήματος και των τεχνικών που χρησιμοποιήθηκαν. Επίσης θα εξηγηθούν διάφορα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη και λειτουργία του συστήματος.

Στο τρίτο κεφάλαιο θα γίνει αναφορά σε σχετική δουλειά που έχει γίνει όσο αφορά την ανάλυση και εκμετάλλευση τηλεπικοινωνιακών δεδομένων για την επίτευξη διάφορων σκοπών όπως η μεγιστοποίηση κερδών των παροχών και η βελτίωση του δικτύου. Θα γίνει επίσης αναφορά στις προσπάθειες που γίνονται από μεγάλους οργανισμούς για την προώθηση της ανάλυσης δεδομένων μέσα σε τηλεπικοινωνιακούς οργανισμούς.

Στο τέταρτο κεφάλαιο θα γίνει αναλυτική περιγραφή της αρχιτεκτονικής του συστήματος και των επιμέρους συστατικών που το αποτελούν καθώς επίσης και μια μικρή αναπαράσταση της ροής της πληροφορίας μέχρι να προκύψει κάτι χρήσιμο που μπορεί να προβληθεί στο χρήστη.

Στο πέμπτο κεφάλαιο θα γίνει παρουσίαση και επεξήγηση των τελεστών που υλοποιήθηκαν στα πλαίσια αυτής της διπλωματικής εργασίας.

Στο έκτο κεφάλαιο θα περιγραφούν οι εφαρμογές παρουσίασης και οπτικοποίησης του συστήματος καθώς και η υποδομή που αξιοποιείται για τις εφαρμογές αυτές.

Στη συνέχεια, στο κεφάλαιο 7 θα γίνει μια πειραματική αξιολόγηση του συστήματος και στο χρόνο ανταπόκρισης του.

Στο όγδοο και τελευταίο κεφάλαιο θα παραθέσουμε τα τελικά συμπεράσματα σχετικά με το όλο σύστημα που αναπτύχθηκε καθώς επίσης και μελλοντικές επεκτάσεις και βελτιώσεις που μπορούν να γίνουν.

Κεφάλαιο 2

Γνωσιολογικό Υπόβαθρο

2.1	Αρχιτεκτονική και Συστήματα Τηλεπικοινωνιακών Παροχών	7
2.1.1	Κεραίες Κυψέλης	8
2.2	Δεδομένα Τηλεπικοινωνιακών Παροχών	7
2.2.1	Call Detail Record (CDR)	9
2.2.2	Network Measurement Report (NMR)	10

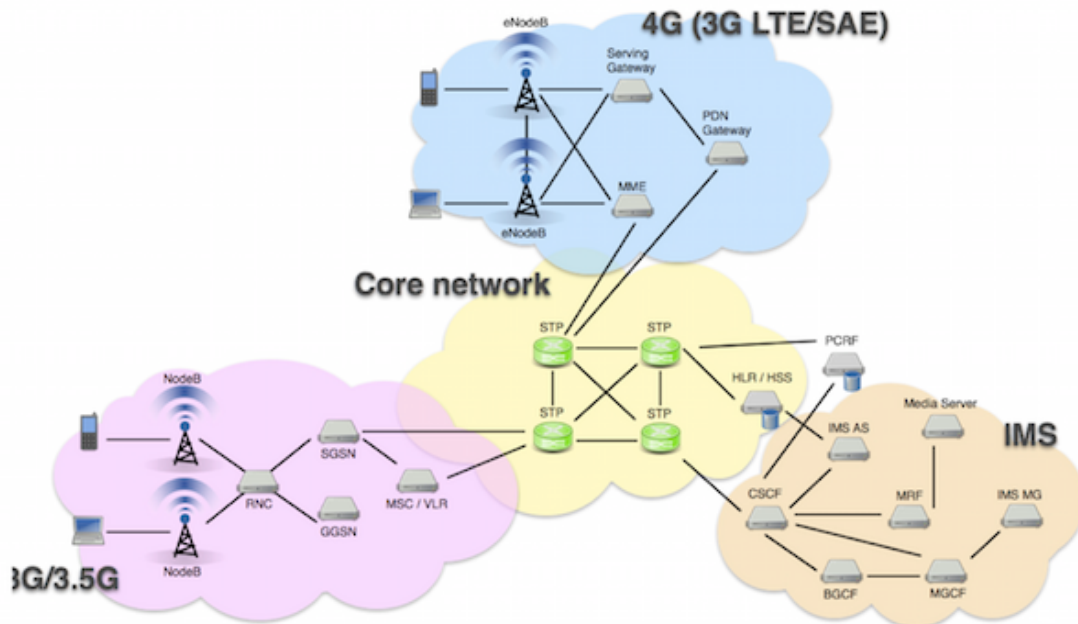
Σε αυτό το κεφάλαιο θα εξηγηθεί πως λειτουργούν τα συστήματα ενός τηλεπικοινωνιακού παροχέα καθώς επίσης και το πώς παράγονται τα δεδομένα. Επίσης, θα εξηγηθούν τηλεπικοινωνιακοί όροι οι οποίοι θα χρησιμοποιούνται εκτεταμένα σε όλη τη διπλωματική εργασία.

2.1 Αρχιτεκτονική και Συστήματα Τηλεπικοινωνιακών Παροχών

Η συζήτηση θα ξεκινήσει με μια αφαιρετική αρχιτεκτονική ενός τηλεπικοινωνιακού παροχέα.

Όπως παρουσιάζεται στο *Σχήμα 2.1* οι τηλεπικοινωνιακοί παροχείς ακολουθούν μια σχετικά απλή αρχιτεκτονική δικτύου σχήματος αστεριού. Πιο συγκεκριμένα στο κέντρο του δικτύου υπάρχει ο πυρήνας που συνδέει όλους τους τύπους δικτύων μεταξύ τους ούτως ώστε να εξασφαλίζεται η συμβατικότητα. Στην αρχιτεκτονική αυτή εμπλέκονται συστατικά τα οποία είναι υπεύθυνα για την καταγραφή διάφορων δεδομένων που ρέουν στο δίκτυο παραδείγματα των οποίων θα δοθούν στην συνέχεια.

Telecom network architecture



Σχήμα 2.1 Τυπική αρχιτεκτονική δικτύου ενός τηλεπικοινωνιακού παροχέα³

2.1.1 Κεραίες Κυψέλης

Οι κινητές συσκευές συνδέονται με τον έξω κόσμο και μεταξύ τους μέσω των διάφορων κεραιών των τηλεπικοινωνιακών παροχέων οι οποίες όπως φαίνεται στο Σχήμα 2.1 συνδέονται μέσω άλλων συστατικών με τον πυρήνα του δικτύου. Η απόσταση από μια κεραία σε άλλη εξαρτάται από πολλούς παράγοντες όπως για παράδειγμα τον μέσο αριθμό των χρηστών που αναμένεται να είναι συνδεδεμένοι σε μια περιοχή. Για την βελτίωση του δικτύου αλλά επίσης και για λόγους χρέωσης των πελατών τους, οι παροχείς μαζεύουν και αποθηκεύουν πληροφορίες και μετρήσεις σχετικά με τον φόρτο κάθε κεραίας αλλά και για τις δραστηριότητες που εκτέλεσε κάποιος χρήστης μέσω μιας κεραίας.

Καθώς κάποιος χρήστης κινείται στο χώρο, αν βγει εκτός εμβέλειας μιας κεραίας τότε η συσκευή είναι υπεύθυνη να ενωθεί απευθείας με κάποια άλλη. Η ενέργεια αυτή ονομάζεται handover και συμβαίνει τόσο γρήγορα που δεν μπορούμε να το

³ <http://labs.plsec.com/2013/07/11/rmll-lsm-2013-opening-up-mobile-and-telecommunications-networks/>

αντιληφθούμε. Εντούτοις το σύστημα έχει την δυνατότητα να δρομολογήσει τα πακέτα φωνής μέσω της νέας κεραίας χωρίς να δημιουργήσει κάποιο κενό στην υπηρεσία.

2.2 Δεδομένα Τηλεπικοινωνιακών Παροχών

Σε αυτή την ενότητα θα γίνει εκτενής περιγραφή και επεξήγηση όλων δεδομένων που ρέουν στο σύστημα ενός τηλεπικοινωνιακού παροχέα καθώς και ο τρόπος με τον οποίο παράγονται.

2.2.1 Call Detail Record (CDR)

SerialNo	CallingParty	CalledParty	CellId	...	StartDate
12398123923	99123456	96987654	2801013125123		20150504132410

Σχήμα 2.2 Παράδειγμα δομής ενός Call Detail Record όπως αυτό προκύπτει από τα συστήματα των παροχών.

Με κάθε δραστηριότητα του χρήστη όπως για παράδειγμα φωνητικές κλήσεις, γραπτά μηνύματα ή σύνδεση στο διαδίκτυο δημιουργείται μια εγγραφή, ονομαζόμενη CDR, στο σύστημα του τηλεπικοινωνιακού παροχέα με δομή όπως αυτή που φαίνεται στο Σχήμα 2.2. Η εγγραφή αυτή περιλαμβάνει όλες τις λεπτομέρειες της δραστηριότητας που έχει προκύψει. Συγκεκριμένα, η εγγραφή περιλαμβάνει μια μοναδική ταυτότητα, την ταυτότητα της κεραίας στην οποία ήταν συνδεδεμένος ο χρήστης όταν ξεκίνησε τη δραστηριότητα, την διάρκεια, τον παραλήπτη αν πρόκειται για κλήση ή γραπτό μήνυμα, την χρονική στιγμή που ξεκίνησε και πολλά άλλα. Οι παροχείς συγκεντρώνουν όλες αυτές τις εγγραφές περιοδικά και τις χρησιμοποιούν προς υπολογισμό των χρεώσεων κάθε χρήστη. Είναι σημαντικό να αναφερθεί ότι οι εγγραφές αυτές αποτελούν μετα-δεδομένα των ενεργειών των χρηστών και όχι τα δεδομένα καθ' αυτό. Για παράδειγμα, μια εγγραφή που αφορά κάποιο γραπτό μήνυμα δεν περιέχει το κείμενο του μηνύματος αλλά μόνο πληροφορίες γι' αυτό.

Επιπρόσθετα, οι εγγραφές αυτές μπορούν να χρησιμοποιηθούν επίσης από αστυνομικά μέλη για τον περιορισμό της ακτίνας την περιοχής στην οποία βρίσκετε κάποιο άτομο αλλά επίσης και στην αναγνώριση ατόμων που διέπραξαν κάποιο έγκλημα.

2.2.2 Network Measurement Report (NMR)

Οι κινητές συσκευές περιοδικά παίρνουν μετρήσεις σχετικά με την δύναμη του σήματος (RSSI) από τις κεραίες που εμπίπτουν στην εμβέλεια τους. Αυτές οι μετρήσεις αποστέλλονται πίσω στις κεραίες για να δημιουργηθεί η αναφορά των μετρήσεων. Οι μετρήσεις συμπεριλαμβάνουν στατιστικά που έχουν να κάνουν καθαρά με το δίκτυο και την απόδοση του. Με την ανάλυση αυτών των μετρήσεων, οι παροχείς μπορούν να γνωρίζουν την κατάσταση του δικτύου αλλά και την θέση του χρήστη σε επίπεδο κεραίας. Με χρήση τεχνικών triangulation οι παροχείς μπορούν να αυξήσουν την ακρίβεια της εκτίμησης θέσης του χρήστη. Συγκεκριμένα, τα δεδομένα αυτά περιέχουν μετρικές όπως για παράδειγμα ο αριθμός των handover που γίνονται σε κάθε κεραία, των αριθμό των αποτυχημένων συνδέσεων σε κάθε κεραία κ.α.

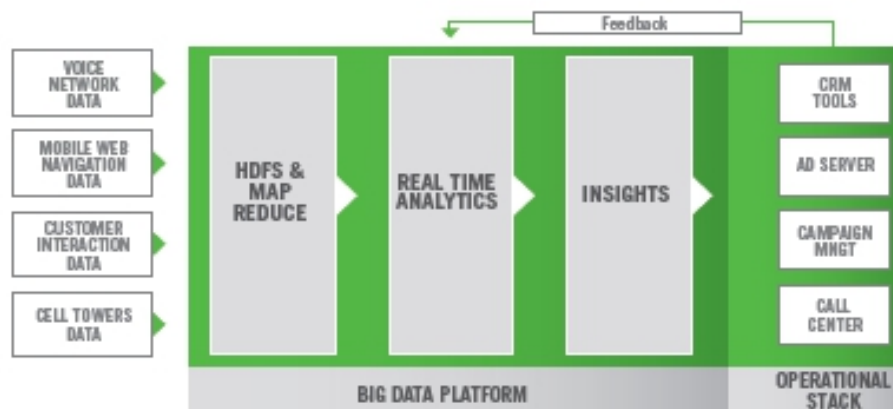
Κεφάλαιο 3

Σχετική Βιβλιογραφία

3.1	Πρόβλεψη των Χρηστών που θα Εγκαταλείψουν την Υπηρεσία	11
3.2	Huawei – Εξειδικευμένη Ανάλυση Μεγάλων Τηλεπικοινωνιακών Δεδομένων	13
3.3	Γεωτοποθέτηση Κινητών Συσκευών στα Τηλεπικοινωνιακά Δίκτυα GSM	15
3.4	OceanRT: Ανάλυση Μεγάλων Χρονικών Δεδομένων σε Πραγματικό Χρόνο	15
3.5	Αναλύοντας Έξι Δισεκατομμύρια CDR την Ημέρα	16
3.6	Data for Development (D4D)	17

3.1 Πρόβλεψη των Χρηστών που θα Εγκαταλείψουν την Υπηρεσία

Οι προσιτές και ανταγωνιστικές τιμές είναι σίγουρα ένας πολύ καλός τρόπος για κάποιο τηλεπικοινωνιακό παροχέα να προσελκύσει νέους πελάτες. Συχνά όμως η διατήρηση τους και ο σχηματισμός μια μακρόχρονης ευτυχισμένης σχέσης μεταξύ πελάτη και παροχέα είναι πολύ δύσκολη. Κάθε καινούργια δελεαστική προσφορά από τους ανταγωνιστές αποτελεί μεγάλο κίνδυνο να χαθούν πελάτες και άρα κίνδυνο για να μειωθούν τα κέρδη. Όμως οι καλές τιμές δεν είναι η μόνη απειλή. Οι πελάτες απαιτούν να έχουν ψηλή ποιότητα και διαθεσιμότητα για τις υπηρεσίες που πληρώνουν και δεν διστάζουν να αλλάξουν παροχέα εάν κάποιος άλλος μπορεί να τους προσφέρει αυτό που επιζητούν. Έτσι είναι σημαντικό για κάποιο παροχέα να προσφέρει τις υπηρεσίες του στην βέλτιστη ποιότητα για να διατηρήσει την πελατεία του ενώ ταυτόχρονα να είναι προσιτός σε νέους πελάτες.



Σχήμα 3.1 Τυπική αρχιτεκτονική ανάλυσης μεγάλων τηλεπικοινωνιακών δεδομένων [1]

Για αυτό το λόγο έχουν αναπτυχθεί τεχνικές και συστήματα για την πρόβλεψη και εντοπισμό πελατών που δεν είναι ευχαριστημένοι με τον τρέχων παροχέα στον οποίο είναι συνδεδεμένοι.

3.1.1 Η Αρχιτεκτονική Συστημάτων Πρόβλεψης Εγκατάλειψης Υπηρεσίας

Αρχιτεκτονικές σχετικά με την πρόβλεψη έχουν μελετηθεί προηγουμένως κυρίως σε ερευνητικό βαθμό [2,1,3]. Ερευνητές από το Universitat Politecnica de Catalunya σε συνεργασία με την Ericsson Ισπανίας σχεδίασαν και ανέπτυξαν μια αρχιτεκτονική για πρόβλεψη των πελατών που έχουν ψηλή πιθανότητα να αλλάξουν παροχέα [4]. Οι ερευνητές υποστήριζαν ότι οι μέχρι τώρα παραδοσιακοί τρόποι πρόβλεψης από την ανάλυση των ιστορικών δεδομένων και την χρήση της μηχανικής μάθησης έχουν περιορισμούς οι οποίοι επηρεάζουν την ακρίβεια της πρόβλεψης. Καταρχάς απαιτούν γνώση όλων των δεδομένων πριν την κατασκευή των μοντέλων πρόγνωσης. Επίσης το γεγονός ότι οι συμπεριφορές των χρηστών αλλάζουν με την πάροδο του χρόνου δεν λαμβάνεται υπόψη από τα μοντέλα που κατασκευάζονται. Αποτέλεσμα αυτών των περιορισμών είναι η ανακρίβεια στις προβλέψεις και η οδήγηση σε λάθος συμπεράσματα.

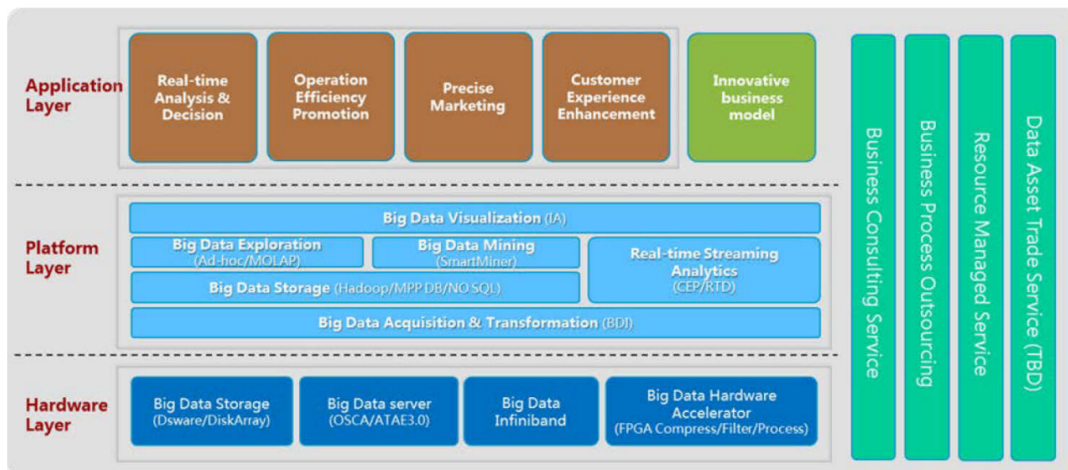
Στη συγκεκριμένη δουλειά προτάθηκε και υλοποιήθηκε μια αρχιτεκτονική η οποία καταναλώνει ζωντανές ροές δεδομένων σε πραγματικό χρόνο και αναπροσαρμόζει τα μοντέλα πρόβλεψης σύμφωνα με τα πιο πρόσφατα μοτίβα που προκύπτουν. Αυτό επιτρέπει στις πιο πρόσφατες συμπεριφορές των χρηστών να λαμβάνουν μεγαλύτερη σημαντικότητα και έτσι το όλο σύστημα να παρέχει αποτελέσματα με μεγαλύτερη ακρίβεια.

Μέσω ενός γραφικού περιβάλλοντος οι ερευνητές μπορούσαν να μεταβάλουν κατά την εκτέλεση παραμέτρους όπως τις τιμές των κλήσεων και τις τιμές των ανταγωνιστών, την συχνότητα των παραπόνων από πελάτες, των μέσο όρο κλήσεων ανά πελάτη κ.α. Κατά την αξιολόγηση του συστήματος παρατηρήθηκε ότι κάθε φορά που οι παράμετροι μεταβάλλονταν, η ακρίβεια του συστήματος έπεφτε αρχικά αλλά μετέπειτα προσαρμοζόταν στα νέα χαρακτηριστικά.

3.2 Huawei – Εξειδικευμένη Ανάλυση Μεγάλων Τηλεπικοινωνιακών Δεδομένων

Είναι γεγονός ότι κάθε τηλεπικοινωνιακός παροχέας ανταγωνίζεται με όλους τους άλλους με σκοπό να αποκτήσει όλο και περισσότερους πελάτες αλλά και να κρατήσει τους τρέχοντες. Όλοι εστιάζονται στην μεγιστοποίηση των εσόδων, στην μείωση του αριθμού των πελατών που μεταφέρονται σε άλλο παροχέα και στην ενίσχυση την ποιότητας υπηρεσίας που προσφέρουν για να ευχαριστούν τους πελάτες. Παρατηρώντας τις συμπεριφορές των πελατών, τις τάσεις της τεχνολογίας και την ποιότητα του δικτύου μπορούν ευκολότερα να πάρουν σημαντικές αποφάσεις που θα οδηγήσουν στην επίτευξη των στόχων τους. Ο όγκος όμως των δεδομένων που μαζεύεται συνεχώς είναι τεράστιος και έτσι οι παραδοσιακές μέθοδοι ανάλυσης δεν είναι πλέον αποδοτικές.

Η έρευνα της Huawei [5] έδειξε ότι με την εκμετάλλευση των δεδομένων μέσω εξειδικευμένων συστημάτων οι παροχείς μπορούν να παίρνουν ευκολότερα σημαντικές αποφάσεις όσο αφορά τις επιχειρησιακές στρατηγικές που πρόκειται να ακολουθήσουν λαμβάνοντας υπόψη προβλέψεις που προκύπτουν από την επεξεργασία. Σύμφωνα με τα



Σχήμα 3.2 Η αρχιτεκτονική του συστήματος της Huawei. [5]

αποτελέσματα των ερωτηματολογίων, 72% των παροχών υποστήριξαν ότι μέσα στα επόμενα 3 χρόνια θα επενδύσουν στην επεξεργασία μεγάλων δεδομένων για καλύτερη ανάλυση των κερδών, 65% θα επενδύσουν στην δυναμική κατάταξη των πελατών σε διάφορες ομάδες ανάλογα με τις τηλεπικοινωνιακές τους συμπεριφορές και 68% θα επενδύσουν στο δυναμικό έλεγχο και αποφυγή συμφόρησης του δικτύου.

3.2.1 Η Λύση Ανάλυσης Μεγάλων Δεδομένων της Huawei

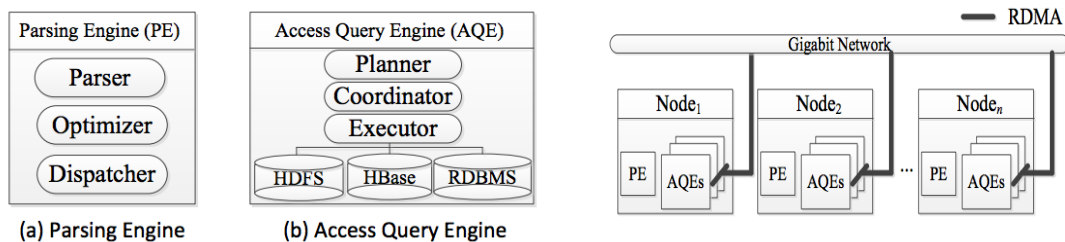
Η δυνατότητα των παροχών να εκμεταλλευτούν στο έπακρο όλη την πληροφορία που έχουν στη διάθεση τους θα ήταν περιορισμένη χωρίς ένα καλά σχεδιασμένο σύστημα επεξεργασίας μεγάλων δεδομένων που να αναλύει και να δημιουργεί σχετικές αναφορές. Έτσι η Huawei κατασκεύασε μια αρχιτεκτονική, όπως αυτή απεικονίζεται στο Σχήμα 3.2, η οποία είναι ικανή να μετατρέπει τα μεγάλα δεδομένα σε χρήσιμη πληροφορία – γνώση βοηθώντας τους παροχείς να λανσάρουν νέες υπηρεσίες πιο προσωπικού χαρακτήρα στους πελάτες τους και να αυξήσουν τα ποσοστά επιτυχίας των καμπάνιων τους. Αν και το σύστημα της Huawei είναι πολύ αποδοτικό και χρησιμοποιείται από πολλούς, εντούτοις παραμένει κλειστό και δεν μπορεί να εξατομικευθεί στις ανάγκες του κάθε τηλεπικοινωνιακού οργανισμού.

3.3 Γεωτοποθέτηση Κινητών Συσκευών στα Τηλεπικοινωνιακά Δίκτυα GSM

Οι εφαρμογές που στηρίζονται ή εκμεταλλεύονται την πληροφορία της θέσης του χρήστη δημιουργούν πλέον ένα νέο τομέα υπηρεσιών ο οποίος όχι μόνο ελκύει τους χρήστες της επόμενης γενιάς αλλά επίσης αποτελεί και μια νέα πηγή εσόδων. Τα είδη τέτοιων εφαρμογών κυμαίνονται από κοινωνικά δίκτυα μέχρι και πλατφόρμες διαφήμισης, από υπηρεσίες επείγουσας ανάγκης μέχρι και υπηρεσίες που παρέχουν πληροφορίες για το τι βρίσκεται στη τρέχουσα θέση του χρήστη. Το πανεπιστήμιο της Μάλτας σε σχετική δημοσίευση [6] παρουσίασε μεθόδους γεωτοποθέτησης των χρηστών σε ένα τηλεπικοινωνιακό δίκτυο οι οποίες δεν απαιτούν οποιοσδήποτε αλλαγές στο ίδιο το δίκτυο ή στις κινητές συσκευές. Οι τεχνικές αυτές βασίζονται κυρίως στην ανάλυση των μετρήσεων σήματος δικτύου (NMR) και τη γεωμετρία.

3.4 OceanRT: Ανάλυση Μεγάλων Χρονικών Δεδομένων σε Πραγματικό Χρόνο

Στα πλαίσια της προσπάθειας για ανάλυση μεγάλων τηλεπικοινωνιακών δεδομένων σε πραγματικό χρόνο, μια ομάδα από ερευνητές στο κέντρο προχωρημένων ψηφιακών ερευνών της Σιγκαπούρης σε σχετική δημοσίευση [7] παρουσίασαν το σύστημα OceanRT. Πρόκειται για ένα σύστημα που βασίζεται σε υποδομή νεφέλης, δηλαδή αξιοποιεί νοητές μηχανές, και σε αντίθεση με άλλα συστήματα που παραδοσιακά κτίζονται πάνω από το πλαίσιο MapReduce, το OceanRT έχει την δική του αρχιτεκτονική και το δικό του αποθηκευτικό σχέδιο. Η αλληλεπίδραση του συστήματος με τον χρήστη γίνεται μέσω επερωτημάτων HiveQL. Η αρχιτεκτονική του περιλαμβάνει διασυνδεδεμένους κόμβους με δίκτυο υψηλής ταχύτητας όπως φαίνεται στο *Σχήμα 3.3*. Ο κάθε κόμβος τρέχει δύο υπηρεσίες, την μηχανή ανάλυσης και βελτιστοποίησης επερωτημάτων και την μηχανή χρονοπρογραμματισμού των επερωτημάτων. Χρησιμοποιώντας ένα ειδικά διαμορφωμένο HDFS το οποίο υποστηρίζει την αποθήκευση των μπλοκ με τέτοιο τρόπο ώστε οι εγγραφές που έχουν παρόμοια χρονικά



Σχήμα 3.3 Τα επίπεδα ανάλυσης ερωτημάτων και της εκτέλεσής τους (αριστερά) και η αρχιτεκτονική του OceanRT (δεξιά) [7]

χαρακτηριστικά να καταλήγουν στο ίδιο μπλοκ ενός αρχείου. Εκμεταλλευόμενο αυτή την διαμόρφωση του HDFS, το OceanRT χτίζει ένα χρονικό ευρετήριο στο κατανεμημένο σύστημα αρχείων πάνω από το οποίο μπορεί να εκτελέσει ερωτήματα σε μετέπειτα χρόνο.

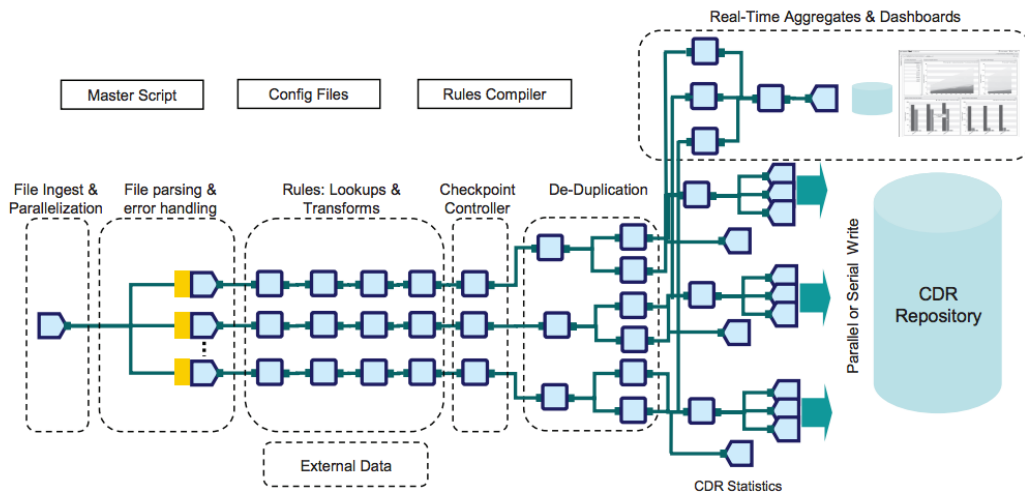
Η πειραματική αποτίμηση του συστήματος απέδειξε την υψηλή απόδοση του κατατάσσοντας το 200 φορές πιο γρήγορο από το Hive⁴ και 12 φορές πιο γρήγορο από το Shark⁵ (πρώιμη έκδοση του Spark SQL). Το σύστημα είναι επεκτάσιμο με την πρόσθεση περισσότερων υπολογιστικών μονάδων όμως δεν υποστηρίζει την εξατομίκευση με την πρόσθεση περισσότερων συναθροιστικών τελεστών ή γραφική παρουσίαση αποτελεσμάτων.

3.5 Αναλύοντας Έξι Δισεκατομμύρια CDR την Ημέρα

Με αφορμή την αξία που προσφέρει η ανάλυση των CDR δεδομένων σε ένα τηλεπικοινωνιακό παροχέα και τον μεγάλο όγκο δεδομένων που αντιμετώπιζε ένας συγκεκριμένος παροχέας, η IBM παρουσίασε μια αρχιτεκτονική το 2012 στο DEBS η οποία μπορεί να επεξεργαστεί σε πραγματικό χρόνο μέχρι και έξι (6) δισεκατομμύρια CDR [8]. Η λύση που σχεδίασαν είναι επεκτάσιμη με την πρόσθεση περισσότερων υπολογιστικών μονάδων. Η αρχιτεκτονική της λύσης φαίνεται στο Σχήμα 3.4.

⁴ Hive Homepage - <https://hive.apache.org/>

⁵ Shark - <http://shark.cs.berkeley.edu/>



Σχήμα 3.4 Η αρχιτεκτονική της λύσης που πρότεινε η IBM για την ανάλυση των δεδομένων σε πραγματικό χρόνο [8]

Το σύστημα καταναλώνει την ροή των CDR, τα αναλύει και παράγει συναθροιστικά στατιστικά. Έπειτα, αποθηκεύει τα CDR δεδομένα σε ένα σύστημα αρχείο κρατώντας έτσι και το ιστορικό τους. Τα συναθροιστικά δεδομένα καταλήγουν στο σύστημα παρουσίασης.

Το σύστημα παρέχει υψηλή απόδοση και παρουσίαση των αποτελεσμάτων όμως προσφέρει μόνο συναθροιστικά αποτελέσματα. Επίσης δεν προσφέρει κάποια προγραμματιστική διεπαφή για την υποστήριξη περισσότερων εφαρμογών παρουσίασης.

3.6 Data for Development (D4D)

Η Orange, ένας τηλεπικοινωνιακός οργανισμός παγκόσμιας εμβέλειας, στην προσπάθεια της να μετατρέψει τα ανοιχτά δεδομένα σε κινητήριο δύναμη για κοινωνικοοικονομική ανάπτυξη, έδωσε ελεύθερη πρόσβαση σε δεδομένα που συλλέχθηκαν από το δικό της δίκτυο στην Ακτή Ελεφαντοστού.

3.6.1 Η Πρόκληση

Για να ενισχύσει ακόμη περισσότερο αυτή την προσπάθεια, η Orange κήρυξε ένα διαγωνισμό⁶. Στα πλαίσια του διαγωνισμού ενθάρρυνε μέλη της επιστημονικής κοινότητας να αναπτύξουν εφευρετικούς τρόπους ανάλυσης των δεδομένων προς όφελος της χώρας, τρόπους προστασίας των προσωπικών δεδομένων των χρηστών αλλά και ευκαιρίες για άλλες προκλήσεις. Ο διαγωνισμός παρουσιάστηκε για πρώτη φορά το 2012 όμως έκτοτε διοργανώνεται κάθε χρόνο. Συγκεκριμένα πέρσι, στα πλαίσια του διαγωνισμού είχαν καθοριστεί 5 θέματα προς αντιμετώπιση. Τα θέματα ήταν σχετικά με την υγεία, την γεωργία, τα μέσα μεταφοράς, την ενέργεια και διάφορα στατιστικά για το έθνος. Τέλος, ο διαγωνισμός στοχεύει να συνεισφέρει και σε ένα πιο τεχνικό επίπεδο βραβεύοντας τις συμμετοχές που καινοτομούν στο τομέα των αλγορίθμων ανωνυμοποίησης, εξόρυξης και παρουσίασης των δεδομένων.

3.6.2 Συστήματα που Προτάθηκαν και Παρουσιάστηκαν

Η μεγάλη ποικιλία των συστημάτων που προτάθηκαν στα πλαίσια του διαγωνισμού είναι η απόδειξη πως τα τηλεπικοινωνιακά δεδομένα μπορούν να εκμεταλλευθούν με πολλούς τρόπους. Η σωστή διαχείριση, επεξεργασία και ανάλυση τους δύναται να βοηθήσει όχι μόνο σε επίπεδα μεγιστοποίησης του κέρδους για τον παροχέα αλλά και ανθρωπιστικό επίπεδο ή επίπεδο βελτίωσης της ποιότητας ζωής.

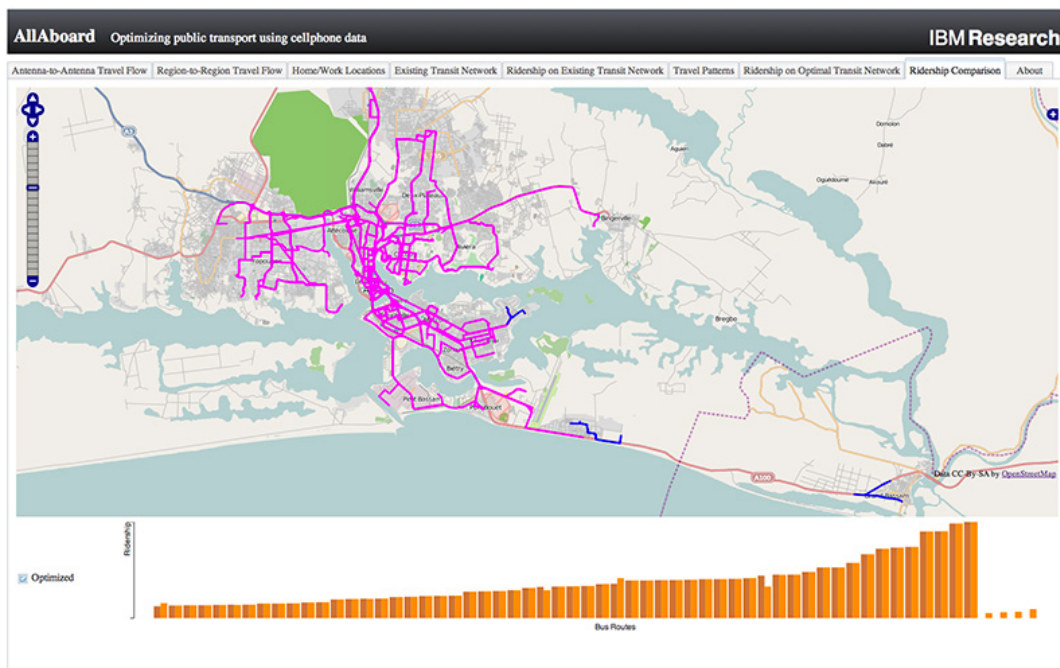
Μερικά από τα έργα που παρουσιάστηκαν στη διάρκεια του περσινού διαγωνισμού είναι τα εξής⁷:

- Ανάλυση ανώνυμων τηλεπικοινωνιακών δεδομένων για να εξετασθεί το που είναι να πιο πιθανό να μαζευτεί ο πληθυσμός μετά από μια φυσική καταστροφή ή επείγουσα ανάγκη. Η πληροφορία αυτή μπορεί μετέπειτα να αξιοποιηθεί για την βελτίωση της κατανομής μονάδων πρώτων βοηθειών σε περιοχές που υπάρχει μεγαλύτερη ανάγκη.

⁶ D4D Challenge - <http://www.d4d.orange.com/en/home>

⁷ D4D Projects - <http://www.orange.com/en/about/our-features/2013/D4D/the-projects>

- Σε περιπτώσεις διοργανώσεων, η κατανόηση του δρομολογίου που ακολουθούν οι επισκέπτες για να παρευρεθούν της διοργάνωσης μπορεί να βοηθήσει στη καλύτερη οργάνωση του και στην αποφυγή συμφόρησης.
- Η συνένωση πληροφοριών κινητικότητας μαζί με πληροφορίες για την εξάπλωση κάποιας ασθένειας βοηθούν στην κατασκευή καλύτερων μοντέλων για τον υπολογισμό της επερχόμενης εξάπλωσης της. Τα μοντέλα αυτά μπορούν μετέπειτα να χρησιμοποιηθούν όταν ξεκινήσει να εξαπλώνεται κάποια επιδημία για να ενημερωθεί ο κόσμος που κινδυνεύει καθώς και για να διοχετευτούν φαρμακευτικές προμήθειες πιο αποδοτικά.
- Η IBM ανέπτυξε ένα σύστημα το οποίο μέσω της εκμετάλλευσης των ανοικτών δεδομένων από την Orange σε συνδυασμό με μερικών δεδομένων για το κυκλοφοριακό δίκτυο, παρήγαγε σε πραγματικό χρόνο ένα χάρτη στον οποίο απεικονιζόταν η κίνηση στους δρόμους. Επίσης, το σύστημα αυτό μπορούσε να χρησιμοποιηθεί για την βελτίωση και δυναμικό σχεδιασμό των δρομολογίων των δημόσιων μέσω μεταφοράς για την καλύτερη εξυπηρέτηση των πελατών. Η γραφική του διεπαφή φαίνεται στο Σχήμα 3.5.



Σχήμα 3.5 Το σύστημα που ανέπτυξε η IBM εκμεταλλευόμενη τα ανοικτά τηλεπικοινωνιακά δεδομένα από την Orange στην Ακτή Ελεφαντοστού⁸

⁸ http://researcher.watson.ibm.com/researcher/view_group_subpage.php?id=4746

Κεφάλαιο 4

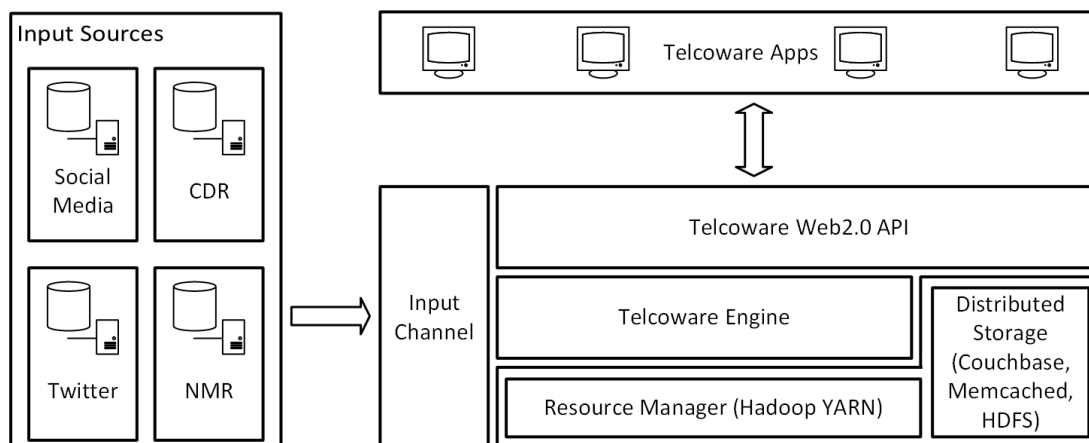
Αρχιτεκτονική του TelcoWare

4.1	Τεχνικό Υπόβαθρο	20
4.2	Υποδομή Ραχοκοκαλιάς	33
4.3	Γεννήτρια Προσομοίωσης Δεδομένων CDR	35
4.4	Συστατικό TelcoWare Master	35
4.5	Εργασίες Επεξεργασίας TelcoWare Jobs	36

Σε αυτό το κεφάλαιο θα γίνει μια εισαγωγή στο τεχνικό υπόβαθρο καθώς και μια επιγραμματική περιγραφή της αρχιτεκτονικής του συστήματος ως ολότητα, όπως αυτό φαίνεται στο *Σχήμα 4.1*. Ακολούθως θα γίνει μια σύντομη περιγραφή του τρόπου χρήσης του συστήματος και της ροής δεδομένων σε αυτό.

4.1 Τεχνικό Υπόβαθρο

Σε αυτή την ενότητα θα αναφερθούν και θα εξηγηθούν όλοι οι απαραίτητοι όροι, αλγόριθμοι και τεχνολογίες που απαιτούνται κατά την επεξήγηση και καλύτερη κατανόηση της αρχιτεκτονικής του συστήματος και του μοντέλου συστήματος που αξιοποιεί.



Σχήμα 4.1 Η αρχιτεκτονική του συστήματος TelcoWare και τα κανάλια επικοινωνίας μεταξύ των συστατικών.

4.1.1 Μεγάλα Δεδομένα και Ροές Δεδομένων (Big Data & Data Streams)

Ο όρος Μεγάλα Δεδομένα αναφέρεται σε σύνολα δεδομένων που έχουν τέτοιο όγκο ή πολυπλοκότητα ώστε τα παραδοσιακά συστήματα επεξεργασίας δεδομένων δεν μπορούν να τα επεξεργαστούν.

Τα μεγάλα δεδομένα παράγονται από τεράστιες εταιρίες με μεγάλο αριθμό χρηστών, οι οποίες συνεχώς παράγουν όλο και περισσότερες πληροφορίες. Ο ρυθμός παραγωγής των δεδομένων αυτών δημιουργεί συνεχώς την ανάγκη για όλο και πιο αποδοτική επεξεργασία τους σε πραγματικό χρόνο. Ο μεγάλος όγκος των δεδομένων που αναφέραμε καθιστά την επεξεργασία τους με συμβατικά εργαλεία αδύνατη. Όμως με σωστές τεχνικές για την επεξεργασία τους, μπορεί κανείς όχι μόνο να κατανοήσει την σημασία τους και να τα αξιοποιήσει την πληροφορία που εμπεριέχουν προς όφελος του αλλά και να προβλέψει μελλοντικές συμπεριφορές.

Το μεγαλύτερο πρόβλημα που προκύπτει από τον όγκο τους είναι το ότι δεν μπορούμε να χωρέσουμε στην μνήμη όλα τα δεδομένα για να προκύψει η επεξεργασία τους. Γι' αυτό και είμαστε αναγκασμένοι να τα επεξεργαζόμαστε σταδιακά, πληρώνοντας μεγάλο κόστος εισόδου εξόδου με τον σκληρό δίσκο, γεγονός που συντείνει στους μεγάλους χρόνους εκτέλεσης.

Για την αντιμετώπιση αυτού το προβλήματος και την αποτελεσματική τους διαχείριση έχουν σχεδιαστεί και αναπτυχθεί κατακευκτωμένα συστήματα, ούτως ώστε ο φόρτος

επεξεργασίας των δεδομένων να μοιράζεται σε πολλούς κόμβους τα οποία θα εξηγηθούν στο επόμενο τμήμα.

Πέρα από τα στατικά μεγάλα σύνολα δεδομένων, σήμερα προκύπτει και η πρόκληση της επεξεργασίας ροών δεδομένων. Ροές δεδομένων είναι συνεχόμενες ακολουθίες δεδομένων που σχεδόν πάντα παράγονται σε πραγματικό χρόνο και οι οποίες μπορούν να καταναλωθούν μόνο μια φορά. Χαρακτηριστικό παράδειγμα είναι η ροή των τιτιβισμάτων του κοινωνικού δικτύου Twitter τα οποία εκμεταλλεύεται η εν λόγω εταιρία για να εντοπίζει διάφορες τάσεις που επικρατούν σε διάφορα σημεία στο κόσμο σε πραγματικό χρόνο.

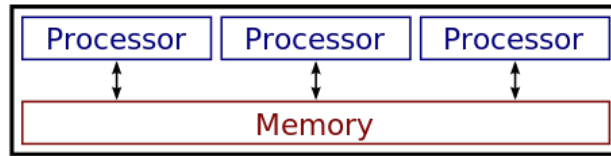
4.1.2 Κατανεμημένη και Παράλληλη Επεξεργασία

Καθώς περνά ο χρόνος, το μέγεθος των δεδομένων που παράγουν οι οργανισμοί αυξάνεται και όπως έχει συζητηθεί στη προηγούμενη ενότητα ο χρόνος απόκρισης των παραδοσιακών συστημάτων ανάλυσης φτάνει σε απαγορευμένα επίπεδα.

Για την αντιμετώπιση και την καλύτερη διαχείριση αυτών των δεδομένων, η ερευνητές στράφηκαν στην Παράλληλη επεξεργασία⁹. Με τον όρο αυτό εννοούμε την διάσπαση του προβλήματος σε επιμέρους, μικρότερα και όπου είναι δυνατόν ανεξάρτητα υπο-προβλήματα τα οποία εκτελούνται ταυτόχρονα (παράλληλα). Έτσι πετυχαίνεται μείωση του χρόνου εκτέλεσης.

Εντούτοις, η έννοια της Παράλληλης Επεξεργασίας, όπως παρουσιάζεται και στο *Σχήμα 4.2*, υπονοεί την ταυτόχρονη χρήση όλων των διαθέσιμων πόρων ενός υπολογιστικού κόμβου και την χρήση κοινής μνήμης για επίτευξη επικοινωνίας. Λόγω του ότι αυτοί οι πόροι δεν είναι άπειροι, η Παράλληλη Επεξεργασία σε ένα και μόνο κόμβο έχει κάποιο μέγιστο όριο βελτίωσης του χρόνου απόκρισης. Προς «αφαίρεση» αυτού του ορίου, αναπτυχθήκαν επίσης τα κατανεμημένα συστήματα επεξεργασίας τα οποία είναι

⁹ Parallel and Distributed Computing - http://en.wikipedia.org/wiki/Distributed_computing



Σχήμα 4.2 Παράλληλη Επεξεργασία σε πολυπύρρηνο επεξεργαστή όπου οι πυρήνες διαμοιράζονται και επικοινωνούν μέσω κοινής μνήμης.

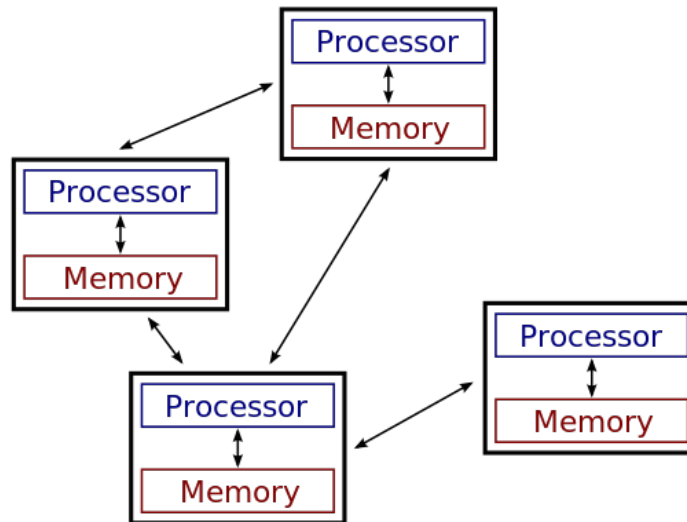
ουσιαστικά συστάδες υπολογιστικών μονάδων που προσφέρουν τους πόρους τους για παράλληλη επεξεργασία.

Τα δύο αυτά μοντέλα, Παράλληλη Επεξεργασία και Κατανεμημένη Επεξεργασία, διαφέρουν ως προς τον τρόπο με τον οποίο επιτυγχάνεται η επικοινωνία μεταξύ των μονάδων επεξεργασίας.

Στα πλαίσια αυτής της διπλωματικής εργασίας μας ενδιαφέρουν περισσότερο τα κατανεμημένα συστήματα επεξεργασίας αφού το πλαίσιο που χρησιμοποιήθηκε και θα εξηγηθεί αργότερα συμπίπτει σε αυτή την κατηγορία.

Κατανεμημένα Συστήματα (Distributed Systems)

Όπως έχει εξηγηθεί συνοπτικά προηγουμένως, κατανεμημένα συστήματα είναι τα συστήματα στα οποία οι υπολογιστικοί κόμβοι δεν είναι κατ' ανάγκη κοντά αλλά απέχουν ο ένας από τον άλλο. Συνήθως οι υπολογιστικοί κόμβοι που αποτελούν το σύστημα απέχουν μερικά μέτρα ή βρίσκονται στο ίδιο κέντρο δεδομένων, όμως σε τέτοια συστήματα μεγάλης κλίμακας οι κόμβοι απέχουν χιλιόμετρα ο ένας από τον άλλο και μπορούν να βρίσκονται μέχρι και σε διαφορετικές ηπείρους. Οι μηχανές που συμμετέχουν στο σύστημα είναι διασυνδεδεμένες σε ένα κοινό δίκτυο μέσω του οποίου γίνεται και η επικοινωνία (κυρίως χρησιμοποιείται το TCP). Πιο συγκεκριμένα, η επικοινωνία πετυχαίνεται μέσω ανταλλαγής μηνυμάτων τα οποία μεταδίδονται μέσω του δικτύου.



Σχήμα 4.3 Υπολογιστικές μονάδες που επικοινωνούν μεταξύ τους μέσω μηνυμάτων. Ο κάθε κόμβος χρησιμοποιεί την δική του μνήμη.

Οι υπολογιστικές μονάδες στα κατακευμασμένα ρυστήματα, όπως φαίνεται στο *Σχήμα 4.3*, δεν διαμοιράζονται κάποια κοινή μνήμη αλλά διατηρούν η κάθε μονάδα την δική της. Για παράδειγμα, εάν κάποιος κόμβος θελήσει να επικοινωνήσει με κάποιον άλλο τότε με την χρήση του πρωτοκόλλου TCP θα στείλει ένα μήνυμα. Λόγω αυτού του μοντέλου αποστολή και παραλαβής μηνυμάτων μπορούμε να πούμε ότι κατά κάποιο τρόπο όλοι οι κόμβοι εφαρμόζουν το μοντέλο πελάτη-εξυπηρετητή (client-server).

Όπως αναφέρθηκε προηγουμένως, το μεγαλύτερο πρόβλημα που αντιμετωπίζεται είναι η έλλειψη κύριας μνήμης. Αν και θα μπορούσαν να κατασκευαστούν υπολογιστικές μονάδες με τεράστια ποσότητα κύριας μνήμης (υπερυπολογιστές), εντούτοις το κόστος κατασκευής τους είναι πολύ μεγάλο. Έτσι οι μεγάλοι οργανισμοί που έχουν την ανάγκη για μεγάλη υπολογιστική ισχύ, στρέφονται στα κατακευμασμένα ρυστήματα που αποτελούνται από μικρότερες, πιο φτηνές υπολογιστικές μονάδες.

Αξίζει να σημειωθεί ότι πλέον υπάρχουν υπηρεσίες όπως το Amazon AWS¹⁰ και το Google Cloud Services¹¹ οι οποίες προσφέρουν την δυνατότητα ενοικίασης υπολογιστικών μονάδων για τον σχηματισμό κατακευμασμένου ρυστήματος χωρίς την ανάγκη συντήρησης.

¹⁰ Amazon AWS Homepage - <http://aws.amazon.com/>

¹¹ Google Cloud Services Homepage - <https://cloud.google.com/>

4.1.3 Εργαλεία Διαχείρισης Μεγάλων Δεδομένων

Για την αποτελεσματική επεξεργασία των μεγάλων δεδομένων απαιτούνται καινούργιες τεχνολογίες και πρωτοποριακοί αλγόριθμοι. Ο μεγάλος τους όγκος καθιστά αδύνατη την επεξεργασία τους από μόνο μια υπολογιστική μονάδα. Στα πλαίσια αυτών των απαιτήσεων έχουν σχεδιαστεί και αναπτυχθεί διάφορα μοντέλα προγραμματισμού τα οποία αποσκοπούν στην πλήρη εκμετάλλευση των κατανεμημένων συστημάτων διαχωρίζοντας τον φόρτο εργασίας και τα δεδομένα σε πολλούς κόμβους. Παραδείγματα τέτοιων πλαισίων είναι το MPI¹², το Apache Hadoop¹³ το οποίο αποτελεί την ανοιχτού κώδικα υλοποίηση του πλαισίου MapReduce¹⁴ της Google, το Apache SparkTM [9], το Apache Storm¹⁵ και διάφορα άλλα.

Στα πλαίσια αυτής της διπλωματικής εργασίας χρησιμοποιήθηκε το σύστημα Apache SparkTM το οποίο περιγράφεται στη συνέχεια.

Apache SparkTM

Το Apache SparkTM είναι μια γρήγορη, ευέλικτη και αφαιρετική μηχανή επεξεργασίας δεδομένων μεγάλης κλίμακας. Το Apache SparkTM έχει σχεδιαστεί και υλοποιηθεί από το ερευνητικό εργαστήριο AMPLab του πανεπιστημίου Berkeley στη Καλιφόρνια. Η διαφορά του από άλλα συστήματα είναι η ικανότητα του να εκτελεί το μεγαλύτερο πιθανό μέρος της επεξεργασίας χρησιμοποιώντας την κύρια μνήμη. Αξιοποιώντας την κύρια μνήμη, μπορεί αποδοτικά να ανταποκριθεί σε μεγάλο φόρτο εργασίας, σε αρκετές περιπτώσεις πολύ πιο γρήγορα από τις εναλλακτικές λύσεις. Το Apache SparkTM μπορεί να λειτουργήσει είτε τοπικά σε ένα υπολογιστικό κόμβο ή κατανεμημένα σε μια συστάδα υπολογιστικών κόμβων.

¹² Message Passing Interface - http://en.wikipedia.org/wiki/Message_Passing_Interface

¹³ Apache Hadoop - <http://hadoop.apache.org>

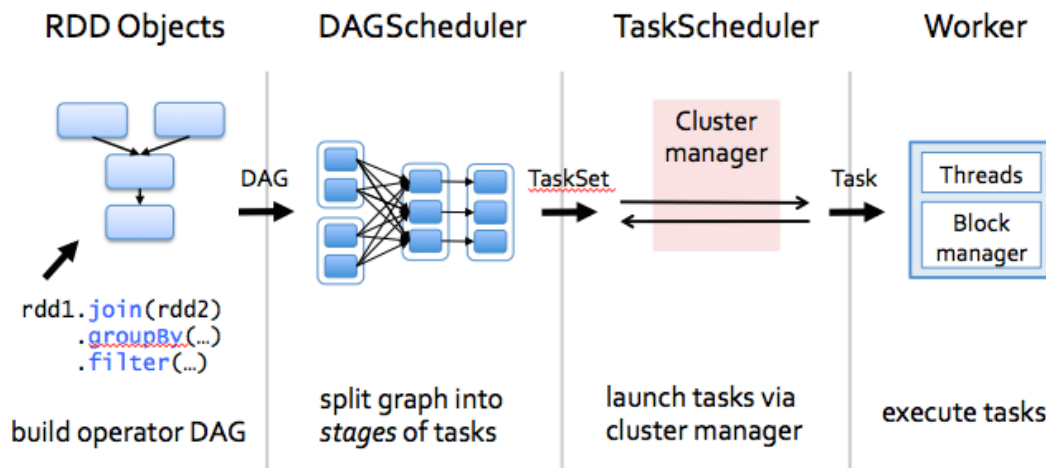
¹⁴ MapReduce Programming Model - <http://research.google.com/archive/mapreduce-osdi04-slides/>

¹⁵ Apache Storm Homepage - <https://storm.apache.org/>

Transformations	$map(f : T \Rightarrow U) : RDD[T] \Rightarrow RDD[U]$ $filter(f : T \Rightarrow Bool) : RDD[T] \Rightarrow RDD[T]$ $flatMap(f : T \Rightarrow Seq[U]) : RDD[T] \Rightarrow RDD[U]$ $sample(fraction : Float) : RDD[T] \Rightarrow RDD[T]$ (Deterministic sampling) $groupByKey() : RDD[(K, V)] \Rightarrow RDD[(K, Seq[V])]$ $reduceByKey(f : (V, V) \Rightarrow V) : RDD[(K, V)] \Rightarrow RDD[(K, V)]$ $union() : (RDD[T], RDD[T]) \Rightarrow RDD[T]$ $join() : (RDD[(K, V)], RDD[(K, W)]) \Rightarrow RDD[(K, (V, W))]$ $cogroup() : (RDD[(K, V)], RDD[(K, W)]) \Rightarrow RDD[(K, (Seq[V], Seq[W]))]$ $crossProduct() : (RDD[T], RDD[U]) \Rightarrow RDD[(T, U)]$ $mapValues(f : V \Rightarrow W) : RDD[(K, V)] \Rightarrow RDD[(K, W)]$ (Preserves partitioning) $sort(c : Comparator[K]) : RDD[(K, V)] \Rightarrow RDD[(K, V)]$ $partitionBy(p : Partitioner[K]) : RDD[(K, V)] \Rightarrow RDD[(K, V)]$
Actions	$count() : RDD[T] \Rightarrow Long$ $collect() : RDD[T] \Rightarrow Seq[T]$ $reduce(f : (T, T) \Rightarrow T) : RDD[T] \Rightarrow T$ $lookup(k : K) : RDD[(K, V)] \Rightarrow Seq[V]$ (On hash/range partitioned RDDs) $save(path : String) : \text{Outputs RDD to a storage system, e.g., HDFS}$

Σχήμα 4.4 Διαθέσιμα transformations και actions στα RDD.

Όλα τα συστήματα επεξεργασίας στο Apache Spark μοιράζονται την ίδια αφαιρετική δομή δεδομένων, την Resilient Distributed Dataset (RDD) [10,11,12,13]. Τα RDD είναι μια κατανεμημένη δομή δεδομένων η οποία επιτρέπει την επεξεργασία με ανοχή σε σφάλματα (fault-tolerant) σε συστάδες υπολογιστικών κόμβων. Προτάθηκαν το 2012 από το πανεπιστήμιο Berkeley της Καλιφόρνιας σαν ιδανική δομή δεδομένων σε δύο (2) περιπτώσεις, των επαναληπτικών αλγορίθμων της μηχανικής μάθησης και των διαδραστικών εργαλείων εξόρυξης δεδομένων, λόγω του ότι επιτρέπουν την εκτέλεση όλων των υπολογισμών στην κύρια μνήμη. Η χρήση της κύριας μνήμης καθιστά τα RDD πολύ πιο αποδοτικά από άλλες τεχνικές που χρησιμοποιούν ενδιάμεσα αρχεία, όπως για παράδειγμα το Hadoop. Στα RDD μπορούν να εκτελεστούν δύο είδη υπολογισμών, τα transformations και τα actions. Τα transformations αφορούν υπολογισμούς από τους οποίους προκύπτει κάποιο καινούργιο RDD και δεν εκτελούνται απευθείας αλλά μόνο όταν ζητηθούν τα αποτελέσματα από κάποιο άλλο υπολογισμό. Ο λόγος για τον οποίο τα transformations δεν υπολογίζονται αμέσως είναι για την βελτιστοποίηση τους η οποία περιλαμβάνει αναδιάταξη ή και συνένωση υπολογισμών. Τα actions αφορούν υπολογισμούς από τους οποίους προκύπτει μια δομή δεδομένων άλλη από τα RDD και σε αντίθεση με τα transformations τα actions υπολογίζονται αμέσως, αναγκάζοντας επίσης και οποιαδήποτε προηγούμενα transformations να υπολογιστούν.



Σχήμα 4.5 Τα επίπεδα που περνά μια εφαρμογή Spark από την υποβολή μέχρι την εκτέλεση.

Σε ψηλό επίπεδο θεωρούμε ότι μια εφαρμογή ή εργασία στο Spark είναι ένα κομμάτι κώδικα το οποίο παίρνει ως είσοδο ένα ή και περισσότερα αρχεία από το HDFS ή από το τοπικό σύστημα διαχείρισης αρχείων, εκτελεί κάποιους είδους επεξεργασία στα συγκεκριμένα δεδομένα και τέλος αποθηκεύει τα αποτελέσματα.

Όλες οι εφαρμογές Spark μπορούν να χωριστούν σε στάδια ανάλογα με τις υπολογιστικές ανάγκες που έχουν. Τα στάδια αυτά μπορούν να κατηγοριοποιηθούν σαν στάδιο Map ή στάδιο Reduce. Κάποιες εφαρμογές μπορούν να εκτελεστούν σε ένα μόνο στάδιο ενώ άλλες χρειάζονται περισσότερα.

Στη συνέχεια, κάθε στάδιο αποτελείται από μικρά έργα. Το έργο αφορά την επεξεργασία ενός μέρους των δεδομένων σε έναν Executor ή αλλιώς η διεργασία που είναι ενεργή σε κάποιο κόμβο και είναι υπεύθυνη για την εκτέλεση του έργου.

Η εκτέλεση εφαρμογών στο Apache Spark βασίζεται στα transformations και actions που καθορίζονται. Κάθε εφαρμογή Spark συμπεριλαμβάνει την διεργασία Driver η οποία είναι υπεύθυνη να εκκινήσει την εφαρμογή και να υποβάλει τους υπολογισμούς στο συγκρότημα υπολογιστικών μονάδων για εκτέλεση. Ο οδηγός αυτός, χρησιμοποιώντας την σειρά και τις εξαρτήσεις μεταξύ των transformations και των actions, χτίζει ένα κατευθυνόμενο μη κυκλικό γράφο (DAG – Directed Acyclic Graph), ο οποίος είναι ουσιαστικά η περίληψη της ροής επεξεργασίας των δεδομένων στην εφαρμογή. Όπως έχει αναφερθεί προηγουμένως, ο γράφος αυτός βελτιστοποιείται από την μηχανή του Spark πριν προχωρήσει στο επόμενο στάδιο. Όταν ο γράφος είναι έτοιμος προωθείται στον επίπεδο του χρονοδρομολογητή γράφων ο οποίος είναι υπεύθυνος να χωρίσει τους

γράφους στα διάφορα στάδια ανάλογα με τις υπολογιστικές τους ανάγκες όπως αναφέρθηκε προηγουμένως. Αφού το κάθε στάδιο διαχωριστεί περαιτέρω σε έργα, αποστέλλεται στο επίπεδο το χρονοδρομολογητή έργων. Το επίπεδο αυτό έχει την ευθύνη να χρονοδρομολογήσει τα έργα στους Workers. Η όλη διαδικασία εκτέλεσης μιας εφαρμογής Spark φαίνεται στο Σχήμα 4.5.

Το Apache Spark σαν σύστημα αξιοποιεί στο έπακρο τα οφέλη της κατανεμημένης επεξεργασίας και προσφέρει διάφορους τρόπους για την υποβολή και εκτέλεση εφαρμογών Spark. Στην έκδοση 1.3.0 η οποία χρησιμοποιείται στα πλαίσια αυτής της διπλωματικής εργασίας υποστηρίζονται οι εξής τρόποι:

- **Τοπική εκτέλεση σε ένα κόμβο (local):** Αυτή η μέθοδος χρησιμοποιείται κυρίως για σκοπούς ανάπτυξης μιας εφαρμογής και για την γρήγορη εκτέλεση μικρών αναλύσεων μέσω της διαδραστικής διεπαφής που προσφέρει το Spark η οποία θα εξηγηθεί πιο μετά. Σε αυτή την μέθοδο, υπάρχει ψευδό-κατανεμημένη επεξεργασία μεταξύ των επεξεργαστών του κόμβου αν χρησιμοποιείται περισσότερο από ένας (1) επεξεργαστής αλλιώς μιλάμε για σειριακό (serial) πρόγραμμα Spark.
- **Standalone Cluster:** Αυτή η μέθοδος παρέχεται από το Spark για αποφυγή χρήσης κάποιου resource manager. Την διαχείριση και χρονοδρομολόγηση των εφαρμογών αναλαμβάνει η διεργασία Spark Master, στην οποία και υποβάλλονται όλες οι εφαρμογές. Όλες οι υπολογιστικές μονάδες στο συγκρότημα τρέχουν την διεργασία Worker, η οποία επικοινωνεί συνεχώς με την διεργασία Master, και προσφέρουν τους πόρους τους για εκτέλεση επιμέρους εργασιών. Η διεργασία Master φροντίζει να δεσμεύει τους πόρους που χρειάζεται μια εφαρμογή για να εκτελεστεί στις υπολογιστικές μονάδες που συμμετέχουν και να τους αποδεσμεύει όταν η εφαρμογή ολοκληρώσει έτσι ώστε να δρομολογηθεί κάποια άλλη. Η μέθοδος αυτή απαιτεί την εγκατάσταση του Spark σε όλους τους κόμβους.
- **Resource Manager Cluster (MESOS, YARN):** Η μέθοδος αυτή αξιοποιεί πλήρως την ιδιότητα κατανομής πόρων που έχουν οι διαχειριστές πόρων και αποφεύγει την χρήση της διεργασίας Spark Master. Έτσι το σύστημα γίνεται πιο

επεκτάσιμο αφού πλέον δεν απαιτείται η εγκατάσταση του Spark σε όλους τους κόμβους και γι' αυτό η πρόσθεση νέων υπολογιστικών μονάδων είναι πολύ ευκολότερη. Παρομοίως με την προηγούμενη μέθοδο, οι εργασίες τώρα υποβάλλονται στο διαχειριστή πόρων ο οποίος με την σειρά του ξεκινά τους Workers στις υπολογιστικές μονάδες που επιλέγονται να συμμετάσχουν στην εκτέλεση.

Παράδειγμα Word Count στο Apache Spark

Για την δημιουργία οποιασδήποτε εφαρμογής Spark είναι απαραίτητη πρώτα η δημιουργία και αρχικοποίηση ενός SparkContext. Το αντικείμενο αυτό είναι υπεύθυνο να δώσει λεπτομέρειες στο οδηγό (Driver) για το πώς να επικοινωνήσει με το συγκρότημα υπολογιστικών μονάδων (cluster) στο οποίο θα εκτελεστεί η εφαρμογή.

Κώδικας 4.1: Δημιουργία πλαισίου SparkContext¹⁶

```
1:  val conf = new SparkConf().setAppName(appName).setMaster(master)
2:  val sc    = new SparkContext(conf)
```

Όπως και σε άλλα συστήματα καταναμημένης επεξεργασίας το πιο ιδανικό παράδειγμα για την επεξήγηση του είναι το πρόβλημα καταμέτρησης των εμφανίσεων κάθε λέξης του κειμένου που βρίσκεται σε ένα ή περισσότερα αρχεία.

Ο αλγόριθμος είναι πολύ απλός. Πρώτα χωρίζουμε τα αρχεία σε λέξεις, δημιουργούμε πλειάδες της μορφής ('λέξη', 1), μαζεύουμε και ομαδοποιούμε τις πλειάδες που έχουν να κάνουν με κάθε λέξη και τέλος αθροίζουμε τους μετρητές.

¹⁶ Spark Programming Guide - <https://spark.apache.org/docs/latest/programming-guide.html>

Κώδικας 4.2: Καταμέτρηση εμφάνισης κάθε λέξης σε κείμενο – Word Count¹⁷

```
1: val conf = new SparkConf().setAppName('Word count').setMaster(master)
2: val sc    = new SparkContext(conf)
3: val textFile = sc.textFile("hdfs://...")
4: val counts = textFile.flatMap(line => line.split(" "))
5:                .map(word => (word,1))
6:                .reduceByKey(_ + _)
7: counts.saveAsTextFile("hdfs://...")
```

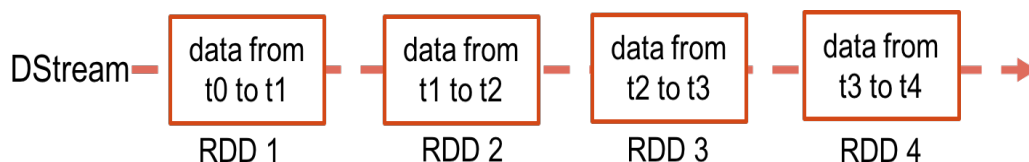
Ο αντίστοιχος κώδικας σε οποιοδήποτε άλλο πλαίσιο επεξεργασίας δεδομένων θα ήταν πολύ περισσότερος και θα περιλάμβανε την δημιουργία διάφορων κλάσεων και συναρτήσεων. Το Apache Spark προσφέρει μια πολύ πιο απλή προσέγγιση στην δήλωση Map-Reduce εργασιών με μια πιο αφαιρετική μορφή. Επίσης προσφέρει την δυνατότητα για συνενώσεις, διασταυρώσεις και ομαδοποιήσεις των δεδομένων αφού αξιοποιεί τα RDD που υποστηρίζουν εξ' ορισμού αυτούς τους τελεστές.

Σύνοψη Προγραμματιστικού Πλαισίου Spark

Μέσω της απλής του προγραμματιστικής διεπαφής, καθιστά πολύ εύκολη την υλοποίηση περίπλοκων αλγορίθμων. Η αφαιρετικότητα του σηκώνει το μεγάλο βάρος από τον προγραμματιστή, ο οποίος δεν χρειάζεται να έχει μεγάλη γνώση κατανεμημένης επεξεργασίας για να υλοποιήσει μια εφαρμογή Spark. Αξιοποιώντας το συναρτησιακό προγραμματιστικό μοντέλο, αφήνει ξεκάθαρη την έννοια του κάθε προγράμματος και αφαιρεί την περιπλοκότητα.

Όμως η απλότητα και αφαιρετικότητα του δεν είναι τα μόνα θετικά. Το Apache Spark έρχεται με επιπλέον βιβλιοθήκες – συστατικά οι οποίες προσφέρουν δυνατότητες για μηχανική μάθηση, εκτέλεση επερωτήσεων σε δεδομένα στη μνήμη, επεξεργασία ροών δεδομένων και επεξεργασία γράφων.

¹⁷ Spark Examples - <https://spark.apache.org/examples.html>



Σχήμα 4.6 Σχηματισμός ενός DStream και τα επιμέρους RDDs που το αποτελούν

Spark Streaming¹⁸

Το Spark Streaming είναι ένα επιπρόσθετο συστατικό-βιβλιοθήκη του Apache SparkTM το οποίο υποστηρίζει την επεξεργασία ροών δεδομένων σε σταθερά χρονικά παράθυρα. Παρέχει την δυνατότητα συνδυασμού ζωντανών δεδομένων (από ροές) μαζί με δεδομένα που βρίσκονται ήδη στη δευτερεύουσα μνήμη. Επίσης, σε περίπτωση κατάρρευσης κάποιου υπολογιστικού κόμβου, είναι ικανό να αποτρέψει τυχόν σφάλματα αναθέτοντας την δουλειά σε κάποιο άλλο κόμβο. Ο εντοπισμός των κόμβων που καταρρέουν γίνεται μέσω αποστολής παλμών. Σε περίπτωση που κάποιος κόμβος αποτύχει να απαντήσει σε κάποιο προκαθορισμένο χρονικό πλαίσιο.

Η κύρια μονάδα δεδομένων στο Spark Streaming είναι το DStream. Στην ουσία το DStream είναι μια σειρά από RDDs τα οποία σχηματίζονται από την εισερχόμενη ροή ανάλογα με το μέγεθος του παραθύρου που έχει καθοριστεί. Κάθε RDD που σχηματίζεται είναι ένα batch δεδομένων το οποίο θα υποστεί την ανάλογη επεξεργασία. Άρα, καθώς η ροή καταναλώνεται από την εφαρμογή, δημιουργείται μια ροή από RDDs (όπως φαίνεται στο Σχήμα 4.6) και αφού γίνει επεξεργασία προκύπτει μια ροή από αποτελέσματα καθώς αυτά αλλάζουν στη πάροδο του χρόνου.

Τέλος, χρησιμοποιώντας το Spark Streaming μπορεί κανείς να αναπτύξει γρήγορα και εύκολα απλές εφαρμογές για επεξεργασία ροών μεγάλων δεδομένων αφού η ίδια η βιβλιοθήκη προσφέρει πολλούς έτοιμους τελεστές ψηλού επιπέδου.

¹⁸ Spark Streaming - <http://spark.apache.org/streaming>

Spark SQL¹⁹

Το Spark SQL είναι μια βιβλιοθήκη του Apache SparkTM η οποία καθιστά εφικτή την επεξεργασία δομημένων δεδομένων. Με την χρήση αυτής της βιβλιοθήκης είναι δυνατή η φόρτωση δεδομένων από διάφορες πηγές δίνοντας τους κάποιο σχήμα και η μετέπειτα εκτέλεση επερωτήσεων πάνω σε αυτά υπό την μορφή SQL.

Για την χρήση του Spark SQL είναι απαραίτητη η αρχικοποίηση ενός SQLContext αντικειμένου ή ενός αντικειμένου που το κληρονομεί. Αυτό είναι υπεύθυνο για να εξηγήσει στο πλαίσιο πώς μπορεί να διαβάσει τα δεδομένα από την πηγή που καθορίζεται. Για παράδειγμα αν τα δεδομένα βρίσκονται στο Apache HIVE τότε είναι απαραίτητη η δημιουργία ενός HiveContext.

Η χαμηλότερη μορφή δεδομένων σε αυτό το πλαίσιο είναι τα DataFrames. Το DataFrame απεικονίζει μια κατανεμημένη συλλογή δεδομένων η οποία χωρίζεται σε ονομασμένες στήλες, κάτι αντίστοιχο δηλαδή μιας σχέσης σε μια παραδοσιακή βάση δεδομένων.

Με την χρήση των DataFrames στα πλαίσια ενός SQLContext επιτρέπεται η εκτέλεση επερωτημάτων κατευθείαν στα δομημένα δεδομένα.

Κώδικας 4.3: Δημιουργία πλαισίου SQLContext και επερωτήματος

```
1:  val sqlContext = ...  // An existing SQLContext
2:  val df = sqlContext.sql("SELECT * FROM table")
```

Όπως φαίνεται στο παράδειγμα *Κώδικας 4.3* η δημιουργία και χρήση ενός SQLContext είναι αρκετά απλή. Με το Spark SQL υπάρχει επίσης η δυνατότητα αποθήκευσης δεδομένων στη δευτερεύουσα μνήμη σε μορφή πίνακα και η επαναφόρτωση τους σε κάποιο δεύτερο χρόνο.

¹⁹ Spark SQL - <http://spark.apache.org/sql/>

Κώδικας 4.4: Αποθήκευση και επαναφόρτωση δεδομένων στο Spark SQL

```
1:  val df = sqlContext.load("people.json", "json")
2:  df.select("name", "age").save("people.parquet", "parquet")
3:
4:  // Read in the parquet file created above. Parquet files are
5:  // self-describing so the schema is preserved.
6:  // The result of loading a Parquet file is also a DataFrame.
7:  val people = sqlContext.parquetFile("people.parquet")
8:
9:  // Parquet files can also be registered as tables and then used
10: // in SQL statements.
11: people.registerTempTable("people")
12: val teens = sqlContext.sql("SELECT name FROM people WHERE age >= 13 AND
13: age <= 19")
14: teens.map(t => "Name: " + t(0)).collect().foreach(println)
```

Το Spark SQL παρέχει όλη την λειτουργία που θα μπορούσε αλλιώς να παρέχει και μια παραδοσιακή σχεσιακή βάση δεδομένων. Όμως, σε αντίθεση με αυτές, το Spark SQL πετυχαίνει το ζητούμενο σε κατανεμημένο περιβάλλον πράγμα το οποίο βελτιώνει την απόδοση και επιτρέπει την εκτέλεση επερωτημάτων σε πολύ μεγαλύτερα σύνολα δεδομένων.

4.2 Υποδομή Ραχοκοκαλιάς

Για να εγγυηθεί η ψηλή επίδοση και εύκολη επεκτασιμότητα του συστήματος TelcoWare έχουν αξιοποιηθεί και συνδυαστεί διάφορες τεχνολογίες και προγραμματιστικά πλαίσια στην ανάπτυξη της ραχοκοκαλιάς.

Η διαδικτυακή επαφή η οποία λαμβάνει όλα τα αιτήματα για παροχή πληροφοριών υποστηρίζεται από το πλαίσιο προγραμματισμού “Play!”²⁰. Όλη η επικοινωνία από έξω προς το σύστημα περνά από την διαδικτυακή επαφή η οποία είναι υπεύθυνη είτε να

²⁰ Play! Framework Homepage – <http://www.playframework.com>

ανακτήσει την απάντηση του ερωτήματος από την βάση δεδομένων (Couchbase Server²¹), είτε να ζητήσει δεδομένα από κάποιο άλλο συστατικό του συστήματος. Η κύρια διεργασία του συστήματος, το συστατικό TelcoWare Master, λειτουργεί αυτόνομα στο σύστημα και επικοινωνεί με τα υπόλοιπα επιμέρους συστατικά χρησιμοποιώντας την τεχνολογία των Akka Actors²².

Για την αποδοτική και επεκτάσιμη λειτουργία των διάφορων εργασιών επεξεργασίας χρησιμοποιήθηκε το πλαίσιο διαχείρισης πόρων Hadoop YARN²³ σε μια συστάδα υπολογιστικών κόμβων. Οι εργασίες επεξεργασίας κάνουν επίσης χρήση του πλαισίου κατανεμημένης επεξεργασίας Apache Spark (βλ. 4.1.3) και αναφέρουν τα αποτελέσματα τους στην βάση δεδομένων.

Στο σύστημα ρέουν 2 κατηγορίες δεδομένων:

1. Δεδομένα με προκαθορισμένη δομή σε μορφή JSON²⁴

Τα δεδομένα που συμπίπτουν σ' αυτή την κατηγορία είναι αποτελέσματα από τις διάφορες εργασίες επεξεργασίας και αποθηκεύονται στη βάση δεδομένων.

2. Μηνύματα που ανταλλάσσονται μεταξύ των Actors (micro-services)

Αν και είναι προσωρινά δεδομένα (δεν αποθηκεύονται πουθενά), εντούτοις ρέουν στο σύστημα της ραχοκοκαλιάς σαν μια μορφή επικοινωνίας μεταξύ των συστατικών.

Όλες οι ιδεατές υπολογιστικές μονάδες της ραχοκοκαλιάς λειτουργούν με Ubuntu 14.04 LTS, 20GB αποθηκευτικό χώρο, 4GB κύρια μνήμη και είναι συνδεδεμένες μεταξύ τους με δίκτυο ψηλής ταχύτητας Gigabit.

²¹ Couchbase Server Homepage – <http://www.couchbase.com>

²² Akka Homepage – <http://www.akka.io>

²³ Hadoop YARN - <http://hortonworks.com/hadoop/yarn/>

²⁴ JSON Homepage – <http://www.json.org>

4.3 Γεννήτρια Προσομοίωσης Δεδομένων CDR

Για την γρηγορότερη ανάπτυξη και ευκολότερη πειραματική αποτίμηση του συστήματος σχεδιάστηκε και υλοποιήθηκε μια γεννήτρια δεδομένων CDR. Συγκεκριμένα, το συστατικό αυτό προσομοιώνει την επικοινωνία μεταξύ 100000 ατόμων και παράγει τις εγγραφές CDR που παραδοσιακά θα δημιουργούνταν από τις κεραιές κάποιου τηλεπικοινωνιακού παροχέα. Στη συνέχεια, η ροή αυτών των εγγραφών καταναλώνεται από της διάφορες εργασίες επεξεργασίας για ανάλυση και εξαγωγή αποτελεσμάτων.

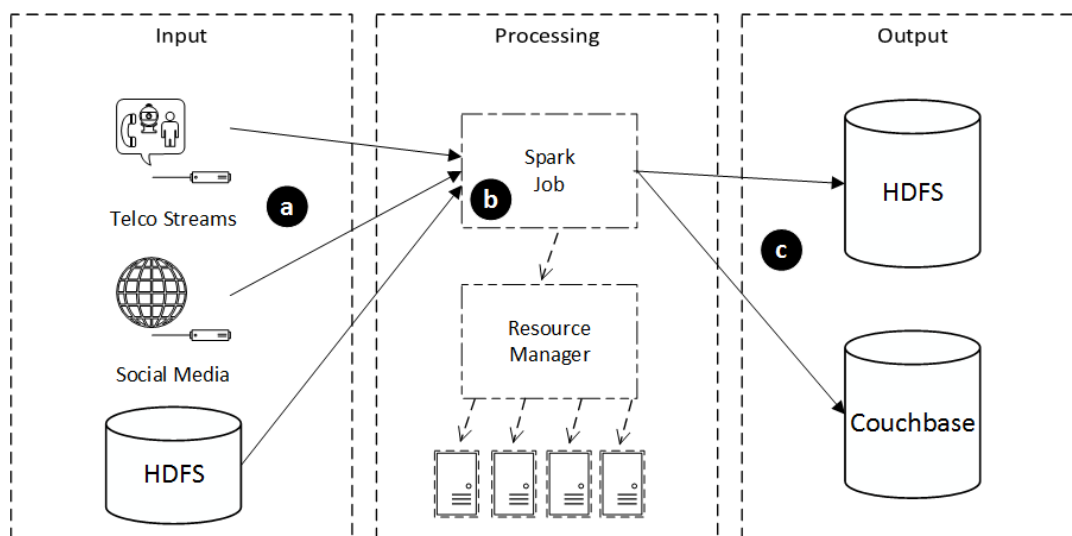
Για να μπορεί να υποστηρίξει πολλές εργασίες ταυτόχρονα, η γεννήτρια είναι κατασκευασμένη ούτως ώστε να εξυπηρετεί κάθε συνδεδεμένο πελάτη σε διαφορετικό νήμα, πετυχαίνοντας έτσι την επιθυμητή παραλληλία.

4.4 Συστατικό TelcoWare Master

Η διεργασία TelcoWare Master ελέγχει περιοδικά την κατάσταση των εργασιών και αναλαμβάνει την ενημέρωση του συστήματος για την ολοκλήρωση ή αποτυχία τους. Βρίσκεται στο κέντρο της αρχιτεκτονικής του συστήματος αφού είναι και υπεύθυνη για την εκκίνηση των κύριων συστατικών του.

Η διεργασία αυτή φροντίζει τον συντονισμό των υπόλοιπων συστατικών και ελέγχει περιοδικά για κατάρρευση κάποιου συστατικού. Σε περίπτωση που εντοπίσει κάποια κατάρρευση, φροντίζει να επανεκκινήσει το συστατικό εάν αυτό ανήκει στο πυρήνα του συστήματος, ή να αναφέρει την αποτυχία στη βάση δεδομένων αν πρόκειται για κάποια εργασία ανάλυσης.

Η διεργασία Master είναι επίσης υπεύθυνη για την δημιουργία μοναδικών ταυτοτήτων για τις εργασίες ανάλυσης αλλά και για την επαλήθευση ταυτότητας του χρήστη που την υποβάλλει.



Σχήμα 4.7 Η ροή πληροφορίας και ο τρόπος λειτουργίας μιας εργασίας ανάλυσης στο σύστημα TelcoWare

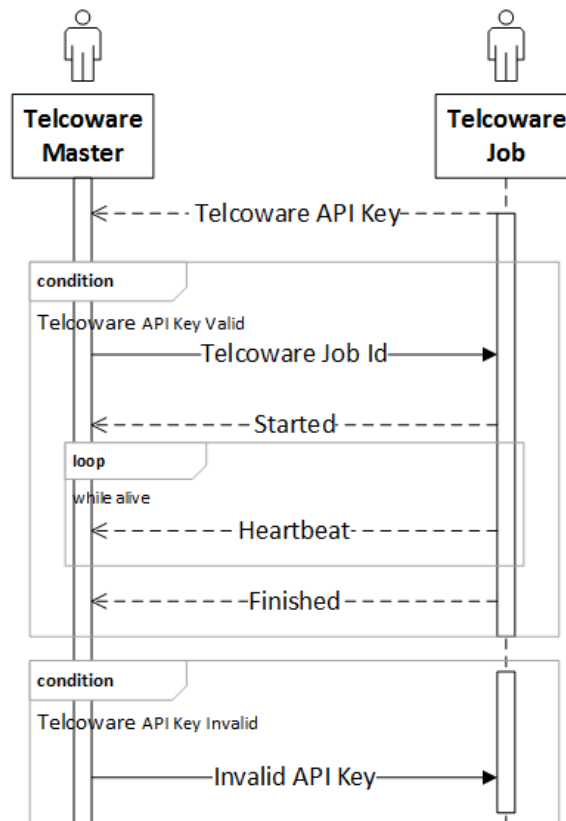
4.5 Εργασίες Επεξεργασίας TelcoWare Jobs

Τα δεδομένα από τις ενέργειες των χρηστών του τηλεπικοινωνιακού παροχέα αλλά και οι μετρήσεις από το δίκτυο του ή και οποιαδήποτε άλλη πηγή από μόνα τους δεν προσφέρουν μεγάλη αξία στον οργανισμό. Περιλαμβάνουν όμως έμμεσα ένα τεράστιο σύνολο πληροφορίας το οποίο μπορεί να αξιοποιήσει ο παροχέας όχι μόνο για να κατανοήσει καλύτερα τις συμπεριφορές και τις ανάγκες των πελατών του αλλά και για να βελτιώσει τις υπηρεσίες που ήδη προσφέρει και να αναγνωρίσει καινούριες υπηρεσίες των οποίων η προώθηση μπορεί να προσθέσει αξία στον οργανισμό.

Για την επίτευξη αυτού του σκοπού τα διάφορα δεδομένα πρέπει να τύχουν επεξεργασίας ανάλογης με αυτή του επιθυμητού αποτελέσματος. Κάθε εργασία επεξεργασίας είναι υπεύθυνη να κάνει μια ή και περισσότερες αναλύσεις στα δεδομένα και να αποθηκεύει τα αποτελέσματα της στη βάση δεδομένων. Το σύστημα μπορεί να υποστηρίξει πολλές ενεργές εργασίες επεξεργασίας ταυτόχρονα ανάλογα με την διαθεσιμότητα των πόρων. Ο «δίκαιος» διαμοιρασμός των πόρων στις TelcoWare Jobs γίνεται μέσω του διαχειριστή πόρων Hadoop YARN. Έτσι εξασφαλίζεται η σωστή και κατανομημένη λειτουργία του συστήματος. Πέραν της επεξεργασίας, οι εργασίες έχουν επίσης την ευθύνη να στέλνουν

μήνυμα παλμού στη κύρια διεργασία TelcoWare Master η οποία κρατά την κατάσταση κάθε εργασίας.

Πριν την εκκίνηση της, μια εργασία ανάλυσης TelcoWare Job, πρέπει να δημιουργήσει ένα TelcowareContext (αντίστοιχο του SparkContext). Το αντικείμενο αυτό έχει την ευθύνη να επικοινωνήσει με το TelcoWare Master αποστέλλοντας του ένα κλειδί. Το κλειδί αυτό είναι απαραίτητο έτσι ώστε το σύστημα να γνωρίζει σε ποιόν ανήκει η εργασία που έχει υποβληθεί αφού συσχετίζεται με ένα χρήστη. Έπειτα, το συστατικό TelcoWare Master, αν το κλειδί είναι έγκυρο, δημιουργεί μια μοναδική ταυτότητα για την εργασία και ανταποκρίνεται με αυτή. Αφού η εργασία παραλάβει την μοναδική της ταυτότητα, μπορεί να ξεκινήσει την εκτέλεση δημιουργώντας το ένα SparkContext όπως έχει εξηγηθεί προηγουμένως. Με σκοπό πάντα την παροχή μιας αφαιρετικής διεπαφής για το χρήστη, η όλη επικοινωνία αφαιρείται από τις ευθύνες του χρήστη αφού παρέχεται μια διεπαφή η οποία ήδη υλοποιεί την επικοινωνία στο παρασκήνιο και αφήνει στο χρήστη την ευθύνη υλοποίησης μόνο της εφαρμογής Spark και του συνόλου των δεδομένων που θα αποθηκευτούν ως αποτελέσματα.



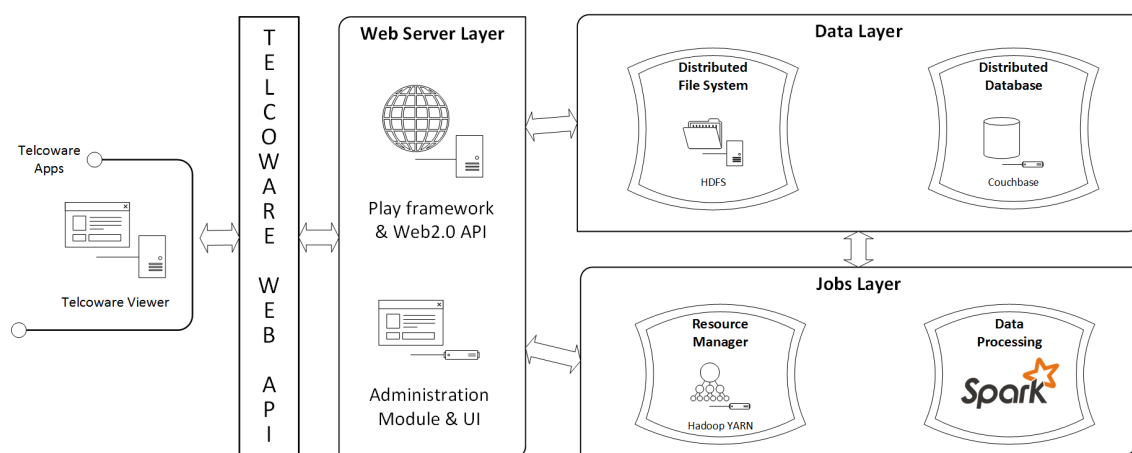
Σχήμα 4.8 Επικοινωνία TelcoWare Job με το συστατικό TelcoWare Master πριν την εκκίνηση της

Κεφάλαιο 5

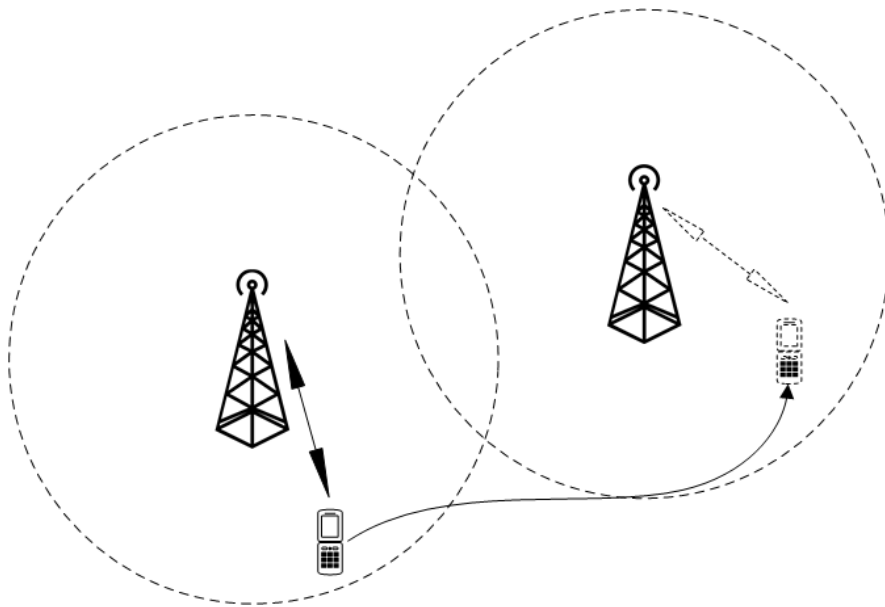
Τελεστές Υψηλού Επιπέδου στο Telcowaare

5.1	Υπολογισμός Συνδεδεμένων Χρηστών ανά Κεραία Κυψέλης	39
5.2	Ευρετήριο Ιστορικού Κλήσεων για Κάθε Χρήστη	42
5.3	Χρονικό Ευρετήριο Χρηστών ανά Περιοχή	43
5.4	Χρονικό Ευρετήριο Αριθμού Χρηστών στο Δίκτυο	45

Η πληροφορία στο TelcoWare ξεκινά από το εσωτερικό σύστημα του παροχέα το οποίο προσφέρει διάφορες ροές δεδομένων. Το TelcoWare βρίσκεται σε θέση να καταναλώνει αυτές τις πηγές και να τις αποθηκεύει υπό μορφή αρχείων σε ένα καταναλωμένο σύστημα διαχείρισης αρχείων. Επίσης, οι ροές αυτές μπορούν να καταναλώνονται από διάφορες ενεργές εργασίες επεξεργασίας. Αφού καταναλωθούν και επεξεργαστούν, τα αποτελέσματα τους αποθηκεύονται σε μια βάση δεδομένων για μετέπειτα ανάκτηση και παρουσίαση. Σε αυτή την ενότητα, θα παρουσιαστούν μερικοί τελεστές οι οποίοι έχουν αναπτυχθεί για σκοπούς δοκιμής και επίδειξης του συστήματος TelcoWare.



Σχήμα 5.1 Επικοινωνία μεταξύ των διάφορων επιπέδων του συστήματος



Σχήμα 5.2 Καθώς οι χρήστες κινούνται στο χώρο συνδέονται με διαφορετικές κεραίες κυψέλης

5.1 Υπολογισμός Συνδεδεμένων Χρηστών ανά Κεραία Κυψέλης

Οι χρήστες του δικτύου μπορούν οποιαδήποτε στιγμή απρόσμενα να προχωρήσουν κάνοντας μια κλήση, στέλνοντας κάποιο γραπτό μήνυμα ή ακόμη και να συνδεθούν στο διαδίκτυο. Μια απότομη αύξηση των χρηστών που είναι συνδεδεμένη σε μια κεραία κυψέλης μπορεί να επιφέρει δραματική πτώση στη ποιότητα υπηρεσίας που λαμβάνουν οι χρήστες. Καθώς κάποιος χρήστης του δικτύου κινείται στο δίκτυο, η σύνδεση του μεταφέρεται από μια κεραία κυψέλης σε μίαν άλλη (Σχήμα 5.2). Η συνεχής επιτήρηση και παρακολούθηση των συνδεδεμένων χρηστών ανά κεραία μπορεί να επιτρέψει την καλύτερη διαχείριση του δικτύου δρομολογώντας τα τηλεφωνήματα ή τα γραπτά μηνύματα των χρηστών με τρόπο έτσι ώστε να αποφεύγεται η συμφόρηση και να αυξάνεται η ποιότητα υπηρεσίας.

Η παραδειγματική για σκοπούς επίδειξης εργασία υπολογισμού συνδεδεμένων χρηστών ανά κεραία δουλεύει σε σχεδόν πραγματικό χρόνο καταναλώνοντας την ροή CDR δεδομένων. Για κάθε εγγραφή που καταναλώνει, εκμεταλλεύεται το πεδίο «Cell Id» για να αυξήσει τον ανάλογο μετρητή. Τα αποτελέσματα των μετρητών ενημερώνονται στην

βάση ανάλογα με το χρονικό παράθυρο που καθορίζεται (στη περίπτωση των επιδείξεων είχε καθοριστεί στα 10 δευτερόλεπτα). Έπειτα, τα αποτελέσματα μεταφέρονται, μέσω του TelcoWare API στην γραφική διεπαφή TelcoWare Viewer η οποία τα παρουσιάζει σε ένα χάρτη.

Κώδικας 5.1: Υπολογισμός συνδεδεμένων χρηστών ανά κεραία – Main method (TelcoWare part)

```
1:  // Δημιουργία αντικειμένου TelcowareConf - επέκταση του SparkConf με
2:  // επιπλέον παραμέτρους που αφορούν το σύστημα TelcoWare
3:  val conf = new TelcowareConf()
4:  conf.setTelcowareMaster("pythia://...")
5:  conf.setAppName("Users per cell tower")
6:
7:  // Δημιουργία του πλαισίου TelcowareContext
8:  val pc = new TelcowareContext(conf, pythiaKey)
9:
10: val job = new CountUsersPerCellTower(cdrHost, cdrPort, 10.seconds, pc)
11:
12: job.start()
```

Το παραπάνω κομμάτι κώδικα υποδεικνύει τον τρόπο με τον οποίο μπορεί να υλοποιηθεί αφαιρετικά μια εργασία ανάλυσης τηλεπικοινωνιακών δεδομένων στο σύστημα TelcoWare. Παρακάτω παρουσιάζεται η δήλωση της ανάλυσης που απαιτείται να γίνει πάνω στα δεδομένα. Για την ανάλυση, όπως είχε προαναφερθεί, αξιοποιείται το πλαίσιο προγραμματισμού Apache SparkTM.

Κώδικας 5.2: CountUsersPerCellTower – Spark

```
1:  // Connect to CDR stream
2:  val recordStrings = streamingContext.socketTextStream(cdrHost, cdrPort)
3:
4:  // Record structure:
5:  //id      |date          |calling   |called    |...|cell id      |...
6:  //21314|20150515005009|35796464126|35796680654|...|280100060151161|...
```

```

7:
8:  // Split the record with character '|'
9:  val records = recordStrings.map(line => line.split("\\|"))
10:
11:  // Field 13 is the cell id - just like word count
12:  val usersPerCell = records.map(rec => (rec(13), 1))
13:                                .reduceByKey((a, b) => a + b)
14:
15:  // Convert text counts to objects
16:  val results = usersPerCell.map(tuple =>
17:                                CellUserCount(tuple._1.toLong, tuple._2))
18:
19:  // Store results to Couchbase
20:  results.foreachRDD(rdd => {
21:    if(rdd.count() > 0) {
22:      points = rdd.collect()
24:      persistData()
25:    }
26:  })
27:
28:  // let's go
29:  streamingContext.start()

```

Στις γραμμές 11 μέχρι 13 στο κώδικα παρατηρείται η χρήση του μοντέλου Map/Reduce για την επίτευξη του στόχου του αλγορίθμου. Από κάθε CDR εγγραφή κρατάμε μόνο το πεδίο 13, την μοναδική ταυτότητα της κεραίας, και δημιουργούμε ζευγάρια κλειδιών-τιμών <‘ταυτότητα κεραίας’, 1>. Έπειτα, για κάθε κλειδί, δηλαδή για κάθε κεραία κυψέλης, συναθροίζονται οι τιμές ούτως ώστε να έχουμε το σύνολο των χρηστών που είναι συνδεδεμένη σε κάθε κεραία. Τέλος, στις γραμμές 19 μέχρι 26 περιγράφεται το ότι για κάθε παρτίδα δεδομένων που επεξεργάζεται, τα αποτελέσματα αποθηκεύονται στη βάση δεδομένων.

Τα παραπάνω κομμάτια κώδικα δείχνουν την απλότητα και μεγάλη αφαιρετικότητα που προσφέρει το σύστημα. Το μόνο σημείο κώδικα για τον συγχρονισμό των δεδομένων στη βάση είναι η γραμμή 24 του *Κώδικας 5.2*. Η σύνδεση με τη διεργασία TelcoWare Master

και την βάση δεδομένων γίνεται στο παρασκήνιο από το περιτύλιγμα που προσφέρει το σύστημα TelcoWare μέσω των βοηθητικών κλάσεων.

5.2 Ευρετήριο Ιστορικού Κλήσεων για Κάθε Χρήστη

Μερικές φορές είναι απαραίτητη η αναγνώριση των ατόμων με τα οποία επικοινωνεί κάποιος. Κτίζοντας ένα ευρετήριο μέσω του οποίου μπορούμε να ανακτήσουμε τα άτομα με τα οποία μιλά κάποιος βοηθά στο να αναγνωρίσει ένας παροχέας ομάδες χρηστών οι οποίοι μιλούν συχνά μεταξύ τους.

Στα άτομα αυτά θα μπορούν να προταθούν πακέτα προσφορών από τον τηλεπικοινωνιακό παροχέα με σκοπό την καλύτερη εξυπηρέτηση του πελάτη. Η βελτίωση αυτής της ποιότητας στην υπηρεσία του παροχέα θα μπορούσε να συντείνει στην μείωση του ποσοστού των πελατών που εγκαταλείπουν την συνδρομή τους για μια άλλη κατά την άποψη τους καλύτερη.

Κώδικας 5.3: Κτίσιμο ευρετηρίου ιστορικού κλήσεων ανά χρήστη – Spark

```
1: case class Call(calling: String, called: String)
2:
3: // Φόρτωση των δεδομένων CDR
4: val historyCDR = sc.textFile(inputPath)
5: val cdrRecords = historyCDR.map(rec => rec.split("\\|"))
6:
7: // Φάση MapReduce για το κτίσιμο του ευρετηρίου calling->called
8: val callIndex = cdrRecords.map(recArray => (recArray(4), recArray(5)))
9:   .reduceByKey((a,b) => a + "," + b)
10:   .map(a => Call(a._1, a._2))
11:
12: callIndex.saveAsTextFile(outputPath)
13: // Φάση MapReduce για το κτίσιμο του αντίστροφου ευρετηρίου
14: // called->calling
15: val inverted = cdrRecords.map(recArray => (recArray(5), recArray(4)))
16:   .reduceByKey((a, b) => a + "," + b)
```

```
17:   .map(a => Call(a._2, a._1))
18:
19:   inverted.saveAsTextFile(outputPath)
```

Για το κτίσιμο του ευρετηρίου η εφαρμογή παίρνει σαν είσοδο τις εγγραφές CDR του παροχέα (γραμμές 4 μέχρι 5). Από κάθε εγγραφή, όπως φαίνεται στη γραμμή 8, επιλέγονται μόνο τα πεδία που αφορούν το αναγνωριστικό (IMSI ή αριθμός τηλεφώνου) του ατόμου που εκτελεί την ενέργεια (calling party) και του ατόμου που λαμβάνει την κλήση ή το γραπτό μήνυμα (called party).

Η καταλληλότερη τεχνική για κατανεμημένη δημιουργία του ευρετηρίου είναι το MapReduce. Στη φάση του Map (γραμμή 8) δημιουργούνται ζευγάρια κλειδιών – τιμών με κλειδί το calling party και τιμή το called party. Αργότερα στη φάση Reduce (γραμμή 9) μαζεύονται όλες οι τιμές για κάθε κλειδί και αποθηκεύονται σε μια ενιαία λίστα.

Αντίστοιχα για το κτίσιμο του αντίστροφου ευρετηρίου, όπως παρουσιάζεται στις γραμμές 15 μέχρι 17, ακολουθείται η ίδια διαδικασία με αυτή του κανονικού ευρετηρίου αντιστρέφοντας τα κλειδιά με τις τιμές όπως χρησιμοποιήθηκαν στο προηγούμενο παράδειγμα.

5.3 Χρονικό Ευρετήριο Χρηστών ανά Περιοχή

Άτομα που επισκέπτονται συχνά μια περιοχή τείνουν να παρουσιάζονται ξανά σε αυτή μέσα σε κάποιο χρονικό διάστημα. Χώροι στους οποίους συχνάζει μεγάλος αριθμός ατόμων μπορούν να εκμεταλλευθούν από τον τηλεπικοινωνιακό παροχέα για λόγους τοποθέτησης διαφημίσεων ή ακόμη και σημείων πρόσβασης στο διαδίκτυο. Επίσης, οι χρήστες που συχνάζουν μαζί σε κάποιες περιοχές ίσως να έχουν και κοινά ενδιαφέροντα. Οι περιοχές στις οποίες συχνάζουν πολλοί χρήστες είναι σίγουρα σημαντικές και κατάλληλες για προώθηση του τηλεπικοινωνιακού παροχέα και των υπηρεσιών του μέσω διαφημίσεων.

Κώδικας 5.4: Εύρεση χρηστών που συχνάζουν στην ίδια περιοχή – Spark

```
1: case class UsersInArea(area: String, users: List[String])
2:
3: // Φόρτωση των δεδομένων CDR
4: val historyCDR = sc.textFile(inputPath)
5: val cdrRecords = historyCDR.map(rec => rec.split("\\|"))
6:
7: // Αποστολή μεταβλητής ορίου εμφανίσεων ενός χρήστη σε μια περιοχή
8: // σε όλους τους κόμβους
9: val _threshold = sc.broadcast(threshold)
10:
11: // Φάση Map/Reduce υπολογισμού εμφανίσεων χρήστη σε μια περιοχή
12: // και ομαδοποίηση χρηστών που εμφανίζονται μαζί
13: val usualLoc = cdrRecords.map(rec => ((rec(4), rec(13)), 1))
14:     .reduceByKey((a, b) => a + b)
15:     .filter(_._2 >= _threshold.value)
16:     .map(r => (r._1._2, List(r._1._1)))
17:     .reduceByKey((a, b) => a ++ b)
18:     .filter(_._2.size >= 2)
19:     .map(pair => UsersInArea(pair._1, pair._2))
20:
21: // Αποθήκευση αποτελεσμάτων
22: usualLoc.saveAsTextFile(outputPath)
```

Αφού φορτωθεί το σύνολο δεδομένων από κάποια πηγή όπως περιγράφεται στη γραμμή 4, τα δεδομένα ετοιμάζονται για επεξεργασία. Έπειτα ακολουθεί η διαδικασία που παρουσιάζεται στις γραμμές 13 μέχρι 19. Αρχικά αξιοποιούνται τα πεδία calling party και cell id για την δημιουργία ζευγαριών κλειδιών-τιμών με την μορφή <<calling_party, cell_id>, 1>. Στη συνέχεια, γίνεται συνάθροιση των τιμών των ζευγαριών που έχουν ίδιο κλειδί. Αφού φιλτραριστούν τα ζευγάρια με τιμή μικρότερη από κάποιο όριο που έχει ορισθεί το οποίο καθορίζει το πόσες φορές χρειάζεται να βρεθεί κάποιος σε μια περιοχή ούτως ώστε η παρουσία του εκεί να θεωρείται συχνή, δημιουργούνται για κάθε κεραία κυψέλης μια λίστα με τους χρήστες που συχνάζουν κάτω από αυτή (γραμμές 17 μέχρι 17). Τέλος, φιλτράρονται οι κεραίες οι οποίες συμπεριλαμβάνουν ομάδες ενός ή δύο

χρηστών αφού είναι ασήμαντες. Τα αποτελέσματα αποθηκεύονται στο κατακευματισμένο σύστημα αρχείων για μετέπειτα ανάκτηση.

5.4 Χρονικό Ευρετήριο Αριθμού Χρηστών στο Δίκτυο

Συχνά, ένας τηλεπικοινωνιακός παροχέας θα ήθελε να γνωρίζει πότε να αναμένει την υπερφόρτωση του δικτύου του ούτως ώστε να λάβει ανάλογα μέτρα και να κρατήσει προσωπικό σε επιφυλακή. Για την επίτευξη αυτού του σκοπού είναι απαραίτητη η αναγνώριση μοτίβων από το ιστορικό των δεδομένων CDR μέσω μηχανικής μάθησης.

Στα πλαίσια της διπλωματικής εργασίας υλοποιήθηκε ένας αλγόριθμος ο οποίος εντοπίζει τις στιγμές στις οποίες υπήρχε η μεγαλύτερη συμφόρηση στο παρελθόν. Τα αποτελέσματα μπορούν να αξιοποιηθούν αργότερα στα πλαίσια της μηχανικής μάθησης για να υπολογιστεί η πιθανότητα να υπάρχει συμφόρηση κάποια χρονική στιγμή στο μέλλον.

Κώδικας 5.5: Κτίσιμο ευρετηρίου ιστορικού κλήσεων ανά χρήστη – Spark

```
1: // Φόρτωση των δεδομένων CDR
2: val historyCDR = sc.textFile(inputPath)
3: val cdrRecords = historyCDR.map(rec => rec.split("\\|"))
4:
5: // Αποστολή μεταβλητής ακρίβειας σε όλους τους κόμβους
6: val suffixLen = sc.broadcast(accuracy)
7:
8: // Φάση Map/Reduce και ταξινόμησης με βάση την ακρίβεια που απαιτείται
9: val actionsCount = cdrRecords.map(rec => {
10:     val len = rec(2).length
11:     (rec(2).substring(0, len - suffixLen.value), 1)
12: }).reduceByKey((a,b) => a + b).sortBy(_._2, ascending = false)
13:
14: // Αποθήκευση αποτελεσμάτων
15: actionsCount.saveAsTextFile(outputPath)
16: if(k.isDefined)
17:     actionsCount.take(k.get).foreach(println)
```

Ανάλογα με την ακρίβεια που απαιτείται, προσαρμόζονται οι χρονοσφραγίδες των εγγραφών CDR, όπως φαίνεται στη 11, και στη συνέχεια χρησιμοποιούνται σαν κλειδί στη φάση Map. Σαν τιμή σε αυτή την φάση χρησιμοποιείται ο αριθμός ένα (1). Στη φάση Reduce γίνεται άθροιση των μονάδων ανά κλειδί για να υπολογιστεί ο αριθμός των χρηστών που ήταν συνδεδεμένος την συγκεκριμένη χρονική στιγμή με την κεραία κυψέλης.

Έπειτα μπορούν να αξιοποιηθούν τεχνικές μηχανικής μάθησης για την εξαγωγή μοτίβων από τη σειρά των χρονικών στιγμών που φορτώνεται το τηλεπικοινωνιακό δίκτυο και την πρόβλεψη μελλοντικών στιγμών στις οποίες ο τηλεπικοινωνιακός παροχέας να αναμένει μεγάλη κίνηση. Αυτό βοηθά στην καλύτερη διαχείριση του δικτύου και την καλύτερη δρομολόγηση των πακέτων επικοινωνίας σε αυτό πετυχαίνοντας εν μέρη βελτίωση στη ποιότητα υπηρεσίας που προσφέρεται.

Κεφάλαιο 6

Εφαρμογές Προβολής και Οπτικοποίησης

6.1	Επίπεδο Εξυπηρετητών	47
6.2	Επίπεδο Δεδομένων	50
6.3	TelcoWare Viewer	52

Το TelcoWare παρέχει μια (1) εφαρμογή Διαδικτύου, την TelcoWare Viewer, μέσω της οποίας απεικονίζονται τα αποτελέσματα των αναλύσεων. Σε αυτό το κεφάλαιο θα παρουσιαστεί αναλυτικά η υποδομή που υποστηρίζει το σύστημα και πιο συγκεκριμένα τα επίπεδα εξυπηρετητών (Web-Server layer) και δεδομένων (Data layer) καθώς και η εφαρμογή που έχει αναπτυχθεί. Τέλος, θα γίνει αναφορά στην εφαρμογή TelcoWare Developers η οποία δίνει ένα σημείο αναφοράς για όλα τα αιτήματα που υποστηρίζει η διεπαφή προγραμματισμού του TelcoWare (Web2.0 API).

6.1 Επίπεδο Εξυπηρετητών

Η υποδομή του TelcoWare αποτελείται από διάφορα συστατικά τα οποία θα εξηγηθούν σε αυτή την ενότητα.

Όλα τα αιτήματα που φτάνουν στους εξυπηρετητές του συστήματος TelcoWare χειρίζονται από το προγραμματιστικό πλαίσιο «Play!» το οποίο ακολουθεί το προγραμματιστικό μοντέλο MVC (Model-View-Controller)²⁵. Το πλαίσιο «Play!» είναι γραμμένο στη γλώσσα προγραμματισμού Scala²⁶ και υποστηρίζει τις γλώσσες Scala και

²⁵ MVC Pattern - <http://en.wikipedia.org/wiki/Model-view-controller>

²⁶ Scala Programming Language - <http://www.scala-lang.org>

Java²⁷. Το συγκεκριμένο πλαίσιο, συνέχισε να αναπτύσσεται μέσα στα χρόνια, απέδειξε την σταθερότητα του και κέρδισε την εμπιστοσύνη μεγάλων οργανισμών όπως για παράδειγμα του LinkedIn²⁸.

Πλαίσιο Εξυπηρετητών (Play Framework)

Το πλαίσιο Play χειρίζεται τα εισερχόμενα αιτήματα χωρίς να διατηρεί την κατάσταση μεταξύ τους (stateless) και επίσης χωρίς να μπλοκάρει σε κάθε αίτημα μέχρι να ολοκληρωθεί πριν προχωρήσει στο επόμενο (non-blocking). Για να πετύχει το δεύτερο, ο χειρισμός των αιτημάτων είναι κτισμένος πάνω από μια ασύγχρονη αρχιτεκτονική, σχετικά καινούργια, αυτή των Akka Actors. Το συγκεκριμένο πλαίσιο είναι εύκολα επεκτάσιμο με την πρόσθεση περισσότερων εξυπηρετητών στη συστάδα αφού η κατάσταση δεν διατηρείται μεταξύ των αιτημάτων.

Λόγω των παραπάνω ιδιοτήτων που διαθέτει το πλαίσιο Play είναι ιδανικό για την εύκολη ανάπτυξη διαδικτυακών διεπαφών προγραμματισμού, ή αλλιώς Web2.0 API. Στα πλαίσια αυτής της διπλωματικής εργασίας χρησιμοποιήθηκε το πλαίσιο Play για την ανάπτυξη του TelcoWare API σε γλώσσα Scala.

Akka Actors

Οι Actors είναι μικρές ελαφριές ταυτόχρονες (concurrent) οντότητες στο σύστημα οι οποίες επεξεργάζονται μηνύματα με ασύγχρονο τρόπο χρησιμοποιώντας βρόγχους παραλαβής γεγονότων (event-driven programming). Με την παραλαβή ενός μηνύματος-γεγονότος, γίνεται αντιστοίχιση του με την ανάλογη ενέργεια που έχει καθορίσει ο χρήστης, αν έχει καθορίσει, η οποία και εκτελείται. Έτσι δημιουργείται μια αφαιρετική δομή ανάπτυξης και συντήρησης ταυτόχρονων συστημάτων που μπορούν να αλληλεπιδρούν μεταξύ τους μέσω των μηνυμάτων. Ως αποτέλεσμα, ο προγραμματιστής μπορεί να συγκεντρωθεί στη ροή των μηνυμάτων στο σύστημα χωρίς να ανησυχεί για λειτουργίες πιο χαμηλού επιπέδου όπως είναι τα νήματα, οι κλειδαριές και η δικτυακή επικοινωνία.

²⁷ Java Programming Language - <https://www.java.com/>

²⁸ Play Framework at LinkedIn - <http://engineering.linkedin.com/play/play-framework-linkedin>

6.1.1 TelcoWare API

Το Web2.0 API του συστήματος έχει αναπτυχθεί για να παρέχει την δυνατότητα σε τρίτους προγραμματιστές (3rd party developers) να έχουν πρόσβαση στα αποτελέσματα των εργασιών επεξεργασίας. Μέσω του TelcoWare API δηλαδή, μπορεί κάποιος να ανακτήσει τα αναλυμένα δεδομένα στα πλαίσια ανάπτυξης κάποιας εφαρμογής σχετικής με την υπηρεσία. Παράδειγμα μιας τέτοιας εφαρμογής είναι το TelcoWare Viewer το οποίο θα εξηγηθεί αργότερα σ' αυτή την ενότητα.

Η προγραμματιστική διεπαφή TelcoWare API υποστηρίζει μηνύματα κειμένου JSON. Όλα τα αιτήματα και όλες οι απαντήσεις φέρουν κάποιο JSON αντικείμενο με όλες τις απαραίτητες πληροφορίες που απαιτούνται από το ανάλογο κάλεσμα.

6.1.2 Συστατικά Εξυπηρετητών (Server Modules)

Πέρα από το Web2.0 API, στο επίπεδο εξυπηρετητών υπάρχουν μερικά ακόμα συστατικά τα οποία φροντίζουν για την ομαλή λειτουργία και συντονισμό του συστήματος και των εργασιών ανάλυσης δεδομένων.

Οι εφαρμογές προβολής και οπτικοποίησης και το API σερβίρονται από τον εξυπηρετητή Netty²⁹ ο οποίος παρέχεται από το πλαίσιο Play σαν συστατικό του. Στο επίπεδο των εξυπηρετητών υπάρχει και το συστατικό TelcoWare Master το οποίο είχε προηγουμένως εξηγηθεί (βλ. 4.4).

Επίσης, στο ίδιο επίπεδο, όπως θα ανέμενε κανείς, λειτουργεί και η διαδικτυακή εφαρμογή TelcoWare Viewer η οποία θα εξηγηθεί και θα αναλυθεί πιο μετά σ' αυτό το κεφάλαιο.

Τέλος, στο επίπεδο εξυπηρετητών, για σκοπούς ανάπτυξης του συστήματος στο ερευνητικό εργαστήριο, λειτουργεί και η γεννήτρια τεχνητών δεδομένων CDR η οποία μπορεί να προσομοιώσει την κίνηση δεδομένων ενός συνόλου κεραιών κυψέλης.

²⁹ Netty Homepage - <http://netty.io/>

6.2 Επίπεδο Δεδομένων

Σε αυτή την ενότητα θα εξηγηθεί το δεύτερο επίπεδο του συστήματος, αυτό των δεδομένων (Data Layer). Το επίπεδο αυτό έχει να κάνει με τον τρόπο που αποθηκεύονται και κινούνται τα δεδομένα στην βάση δεδομένων και στο σύστημα αρχείων.

Στο σύστημα υπάρχουν τρία (3) είδη δεδομένων:

1. Τα αποτελέσματα της ανάλυσης των τηλεπικοινωνιακών δεδομένων
2. Οι μέτα-πληροφορίες των εργασιών ανάλυσης
3. Το ιστορικό των CDR και NMR εγγραφών

Τα δεδομένα αυτά μπορούν να καταταγούν σε δύο (2) διαφορετικές κατηγορίες ανάλογα με τον τρόπο αποθήκευσής τους. Οι κατηγορίες έχουν ως εξής:

1. Δομημένα δεδομένα σε μορφή αντικειμένων JSON (Δεδομένα 1 και 2)
2. Δεδομένα κειμένου τα οποία έχουν κάποια μορφή όμως αποθηκεύονται σαν απλές συμβολοσειρές (Δεδομένα 3)

```
{  "jobId": "job-20150507100404-0002",
  "jobName": "Users per cell tower",
  "userId": "user_shared",
  "isFinished": false,
  "timestamp": 1430982305985
}
```

Σχήμα 6.1 Αναπαράσταση του JSON αντικειμένου με τις πληροφορίες μιας εργασίας ανάλυσης

Το πρώτο είδος δεδομένων αποτελεί τα αποτελέσματα που προκύπτουν από την ανάλυση των διαθέσιμων δεδομένων μέσω του Apache Spark. Αν και τα αποτελέσματα αυτά κατά την επεξεργασία βρίσκονται σε δυαδική μορφή στη κύρια μνήμη, εντούτοις με το πέρας της επεξεργασίας αποθηκεύονται υπό μορφή JSON στην βάση δεδομένων. Λόγω της δομής τους και της γεωγραφικής πληροφορίας που διαθέτουν, σαν βάση δεδομένων για όλο το σύστημα TelcoWare επιλέχθηκε το Couchbase Server. Το Couchbase Server χρησιμοποιεί ως κύρια μονάδα αποθήκευσης του τα JSON αντικείμενα και έτσι είναι ιδανικό για την αποθήκευση των δεδομένων μας. Επίσης, το Couchbase εκμεταλλεύεται

το Memcached³⁰ το οποίο κρατά στην κύρια μνήμη τα πιο πρόσφατα αποτελέσματα και έτσι μπορεί πιο γρήγορα να ανταποκρίνεται σε αιτήσεις που δέχεται επαναλαμβανόμενα. Αυτό αποτελεί ιδανική λύση για τις εφαρμογές που κτίζονται πάνω από το σύστημα.

Το δεύτερο είδος δεδομένων είναι οι μετα-πληροφορίες των εργασιών ανάλυσης. Οι πληροφορίες αυτές μοντελοποιούνται σε JSON αντικείμενα και δηλώνουν την τρέχουσα κατάσταση της εργασίας που αντιπροσωπεύει. Πιο συγκεκριμένα, οι μετα-πληροφορίες που αποθηκεύονται, όπως φαίνεται στο *Σχήμα 6.1*, συμπεριλαμβάνουν μεταξύ άλλων την μοναδική ταυτότητα της εργασίας, το όνομα της, την χρονική στιγμή της τελευταίας ενημέρωσης και το εάν είναι ακόμα ενεργή η εργασία. Όπως και το προηγούμενο είδος δεδομένων, οι μετα-πληροφορίες αποθηκεύονται στην βάση δεδομένων Couchbase Server.

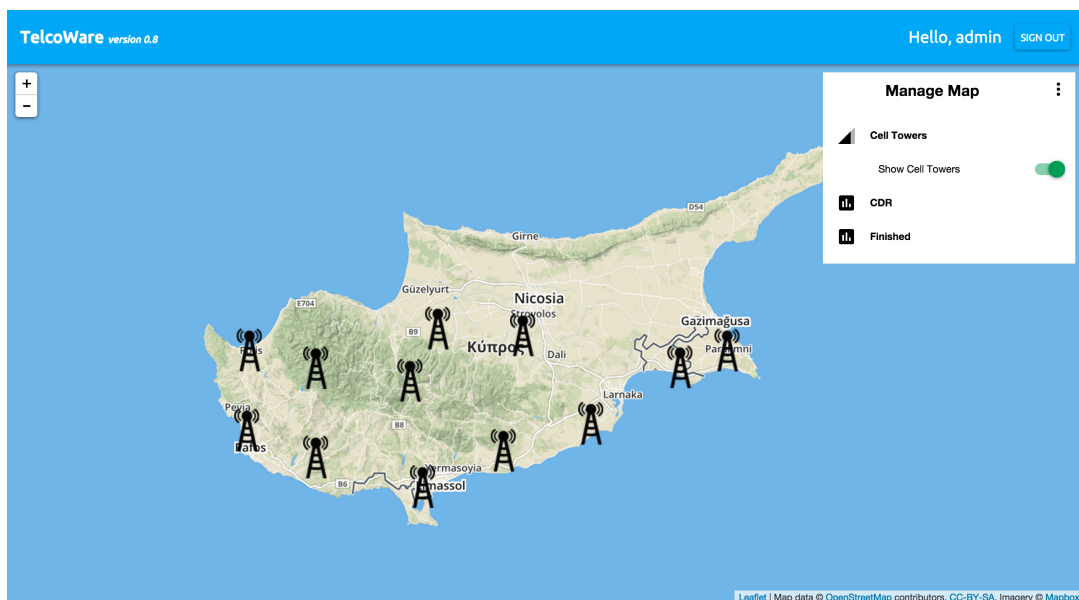
Το τελευταίο είδος δεδομένων διαφέρει από τα προηγούμενα δύο (2) για το λόγο του ότι συμπίπτει σε μεγαλύτερο βαθμό στην κατηγορία των μεγάλων δεδομένων. Τα δεδομένα αυτού του είδους είναι εκείνα που εισέρχονται στο σύστημα από τις διάφορες ροές δεδομένων. Σε περίπτωση που το συστατικό TelcoWare Recorder είναι ενεργοποιημένο, τα συγκεκριμένα δεδομένα δεν καταναλώνονται απλά από τις τρέχων ενεργές εργασίες αλλά και επίσης αποθηκεύονται σε ένα κατανεμημένο σύστημα αρχείων. Μελλοντικές εργασίες μπορούν να αξιοποιήσουν το ιστορικό για σκοπούς όπως εντοπισμό μοτίβων συμπεριφορών ή και προβλέψεις, όπως για παράδειγμα σχετικά με την συμφόρηση του δικτύου. Σαν κατανεμημένο σύστημα αρχείων χρησιμοποιήθηκε το HDFS³¹ αφού το Apache Spark διαθέτει μια πλήρες υλοποιημένη διεπαφή για την διαχείριση αρχείων στο εν λόγω σύστημα. Το HDFS είναι η ανοιχτού κώδικα υλοποίηση του GoogleFS (Google File System) [14], το οποίο προτάθηκε το 2003 στο SOSP.

³⁰ Memcached Homepage - <http://memcached.org>

³¹ HDFS - http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

6.3 TelcoWare Viewer

Λόγω του μεγάλου τους όγκου, τα αποτελέσματα των εργασιών δεν είναι κατανοητά στο ανθρώπινο μάτι. Οι εφαρμογές γραφικής διεπαφής είναι υπεύθυνες να μεταδώσουν την πληροφορία με τρόπο εύκολα και γρήγορα κατανοητό. Για την επίτευξη του σκοπού τους, αξιοποιούν το TelcoWare API για να αντλήσουν τα τελευταία αποτελέσματα που έχουν προκύψει από τις εργασίες επεξεργασίας TelcoWare Jobs. Παράδειγμα μια γραφικής εφαρμογής είναι το TelcoWare Viewer το οποίο θα εξηγηθεί σε αυτή την ενότητα. Η εφαρμογή αυτή είναι προσβάσιμη με την χρήση οποιουδήποτε περιηγητή υποστηρίζει το πρότυπο HTML5.



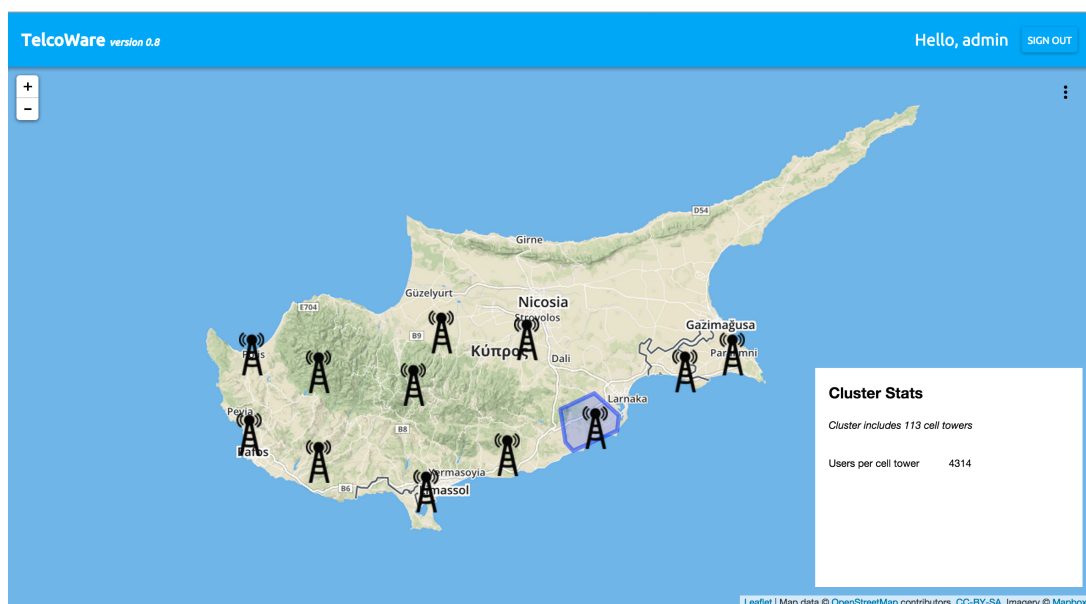
Σχήμα 6.2 Παρουσίαση των ομαδοποιημένων ανά περιοχή κεραιών κυψέλης

Ο Viewer είναι ικανός να παρουσιάσει αποτελέσματα αναλύσεων που υποδηλώνουν στατιστικά για τις κεραιές κυψέλης. Η απλή διεπαφή του Viewer αποτελείται κυρίως από ένα χάρτη και μερικά σημεία επιλογής των αποτελεσμάτων που πρόκειται να παρουσιαστούν. Ο χάρτης που χρησιμοποιείται στην εφαρμογή αποτελείται από το LeafletJS³² ως πυρήνα και παρουσιάζει τα «πλακάκια» (tiles) του γνωστού ανοιχτού συστήματος χαρτογράφησης Open Street Maps³³.

³² LeafletJS Homepage – <http://www.leafletjs.com>

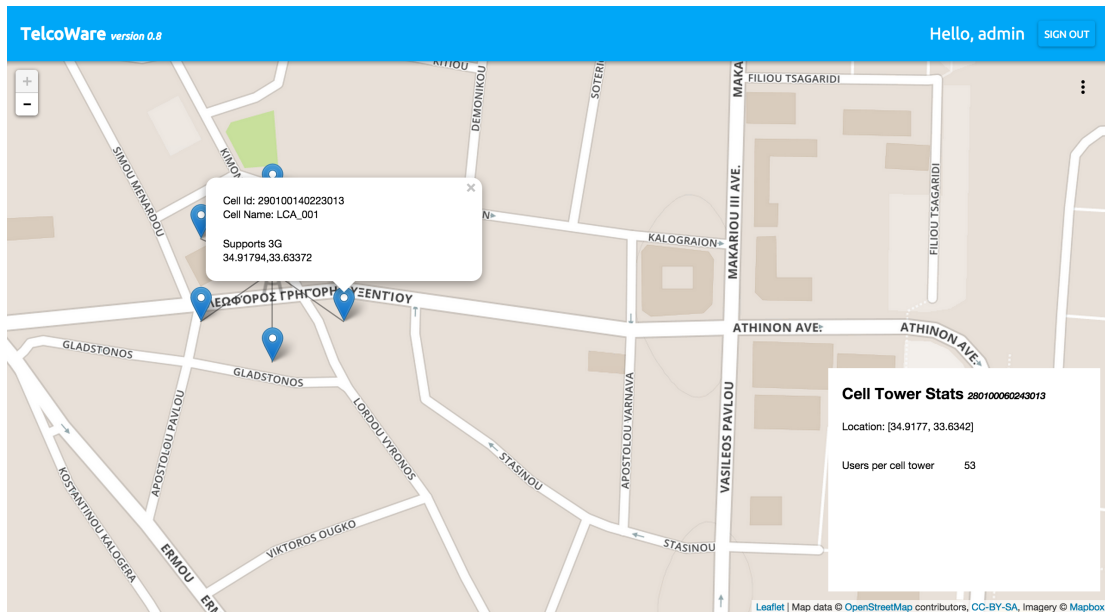
³³ OpenStreetMaps Homepage – <http://www.openstreetmap.org>

Μετά την είσοδο του στο σύστημα, ο χρήστης μεταφέρεται στη σελίδα που παρουσιάζεται στο *Σχήμα 6.2*. Στη συγκεκριμένη σελίδα ο χρήστης μπορεί να δει αναλυτικά τις τοποθεσίες των κεραιών που βρίσκονται στη βάση δεδομένων. Για ευκολία κατανόησης της διεπαφής, η κεραιές κυψέλης ομαδοποιούνται γεωγραφικά ανάλογα με την απόστασή τους. Με την επιλογή μιας ομάδας κεραιών, εμφανίζονται τα σχετικά συναθροιστικά στατιστικά όλων των κεραιών που υπάγονται στη γεωγραφική ομάδα. Η σχετική λειτουργία φαίνεται στο *Σχήμα 6.3*.



Σχήμα 6.3 Παρουσίαση αθροιστικών στατιστικών για μια ομάδα κεραιών κυψέλης από τα αποτελέσματα ανάλυσης

Σε πιο λεπτό επίπεδο, ο χρήστης μπορεί να δει τα στατιστικά μιας και μόνο κεραιάς καθώς και λεπτομέρειες για την συγκεκριμένη κεραιά. Στο επίπεδο της κεραιάς, ο Viewer, όπως απεικονίζεται στο *Σχήμα 6.4*, παρέχει πληροφορίες για την επιλεγμένη κεραιά όπως το Cell Id, το όνομα κυψέλης, τους τύπους δικτύου που υποστηρίζει και το γεωγραφικό σημείο στο οποίο είναι εγκατεστημένη. Επίσης, παρουσιάζονται τα αποτελέσματα των αναλύσεων που σχετίζονται με την συγκεκριμένη κεραιά.



Σχήμα 6.4 Παρουσίαση στατιστικών για μια κεραία κυψέλης καθώς και λεπτομέρειες σχετικά με την συγκεκριμένη κεραία

Κεφάλαιο 7

Πειραματική Αποτίμηση Συστήματος

7.1	Πειραματική Μεθοδολογία	55
7.2	Απόδοση Αναλύσεων	56

Σε αυτό το κεφάλαιο θα γίνει πειραματική αποτίμηση του συστήματος όχι μόνο σχετικά με τον χρόνο εκτέλεσης των εργασιών ανάλυσης αλλά και η απλότητα του σε σχέση με τα παραδοσιακά εργαλεία ανάλυσης δεδομένων που χρησιμοποιούν οι τηλεπικοινωνιακοί παροχείς.

7.1 Πειραματική Μεθοδολογία

Για την αποτίμηση και αξιολόγηση του συστήματος έχουν δημιουργηθεί τρία (3) συνθετικά σύνολα δεδομένων. Το πρώτο σύνολο με μέγεθος περίπου 1GB και συγκεκριμένα 1674770 εγγραφές CDR, το δεύτερο μεγέθους 5GB με 7540977 εγγραφές ενώ το τελευταίο με μέγεθος 10GB και 14868622 εγγραφές CDR. Πιο συγκεκριμένα για την αποτίμηση εκτελέστηκαν οι αλγόριθμοι που είχαν εξηγηθεί στο Κεφάλαιο 5 με τα τρία (3) σύνολα δεδομένων. Επίσης, έγινε μια σύγκριση της υλοποίησης των αλγορίθμων αυτών με τις αντίστοιχες υλοποιήσεις που θα είχε ένας τηλεπικοινωνιακός παροχέας αξιοποιώντας τα παραδοσιακά του εργαλεία ούτως ώστε να αναδειχθεί η αφαίρεση της πολυπλοκότητας.

Τα πειράματα εκτελέστηκαν αξιοποιώντας ένα συγκρότημα 4 υπολογιστικών μονάδων με τα εξής χαρακτηριστικά:

- Ubuntu 14.04
- 2 cores @ 2.0GHz
- 4GB RAM
- 20GB HDD

Επίσης, στα πλαίσια των πειραμάτων, για τις εργασίες ανάλυσης ροών δεδομένων, η γεννήτρια ρυθμίστηκε να παράγει φόρτο εργασίας 10 με 29 MB το δευτερόλεπτο (10~29 MBps). Αν και είναι σχετικά μικρός φόρτος, είναι ικανοποιητικός για την ανάπτυξη και αποσφαλμάτωση του συστήματος και προσομοιώνει σε μεγάλο βαθμό την ροή του τηλεπικοινωνιακού παροχέα στον οποίο έχει εγκατασταθεί.

7.2 Απόδοση Αναλύσεων

Παρακάτω παρουσιάζεται ο χρόνος εκτέλεση των εργασιών που υλοποιήθηκαν καθώς και των αντίστοιχων εργασιών υλοποιημένες με παραδοσιακά εργαλεία τηλεπικοινωνιακών παροχέων όπως Bash³⁴ και Python³⁵.

Αξίζει να σημειωθεί ότι τα παραδοσιακά εργαλεία χρησιμοποιήθηκαν με σειριακό τρόπο χωρίς να υλοποιούν κάποιους είδους παραλληλία. Αντίθετα, οι εργασίες ανάλυσης που υλοποιήθηκαν στο Apache Spark είναι εκ φύσεως κατανεμημένες.

³⁴ Bash Scripting Language - <http://www.gnu.org/software/bash/manual/bashref.html>

³⁵ Python - <https://www.python.org/>

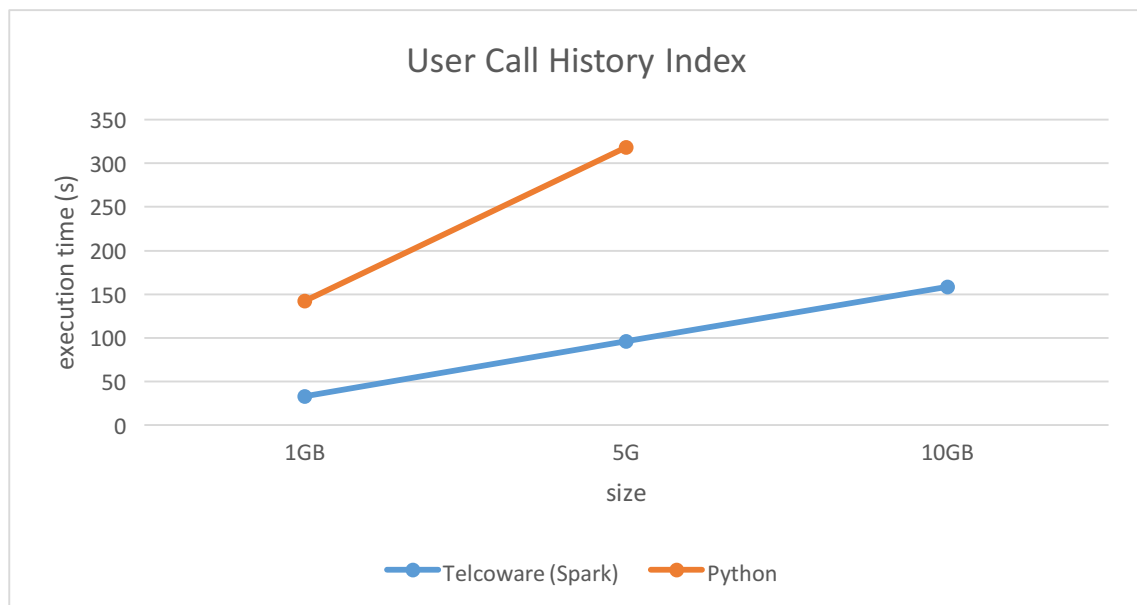
Ζωντανή Καταμέτρηση Συνδεδεμένων Χρηστών ανά Κεραία Κυψέλης

Metric	Minimum	Median	Maximum
Processing Time	5 seconds 93 ms	6 seconds 636 ms	9 seconds 100 ms
Scheduling Delay	0 ms	0 ms	3 ms
Total Delay	5 seconds 94 ms	6 seconds 636 ms	9 seconds 100 ms

Σχήμα 7.1 Χρόνος ανάλυσης της ροής των CDR δεδομένων που πηγάζει από την γεννήτρια

Όπως έχει αναφερθεί προηγουμένως η γεννήτρια δεδομένων για σκοπούς προσομοίωσης ενός πραγματικού περιβάλλοντος είναι ρυθμισμένη να παράγει φόρτο εργασίας από 10 μέχρι και 29 MB το δευτερόλεπτο (10-29 MBps). Το χρονικό παράθυρο στην εργασία επεξεργασία έχει καθοριστεί στα δέκα (10) δευτερόλεπτα ούτως ώστε το σύστημα να έχει το περιθώριο να ανταποκριθεί σε περίπτωση που υπάρξει μια έκρηξη χρήσης δικτύου. Παρατηρείται από το Σχήμα 7.1 ότι ο ελάχιστος χρόνος που χρειάστηκε για την επεξεργασία μιας παρτίδας δεδομένων ήταν περίπου πέντε (5) δευτερόλεπτα, ενώ ο μέγιστος έφτασε τα εννέα (9) δευτερόλεπτα. Αυτό συμβαίνει λόγω του ότι δεν έχουν όλες οι παρτίδες το ίδιο μέγεθος σε δεδομένα αφού η ροή εισόδου δεν διατηρεί κάποιο σταθερό ρυθμό.

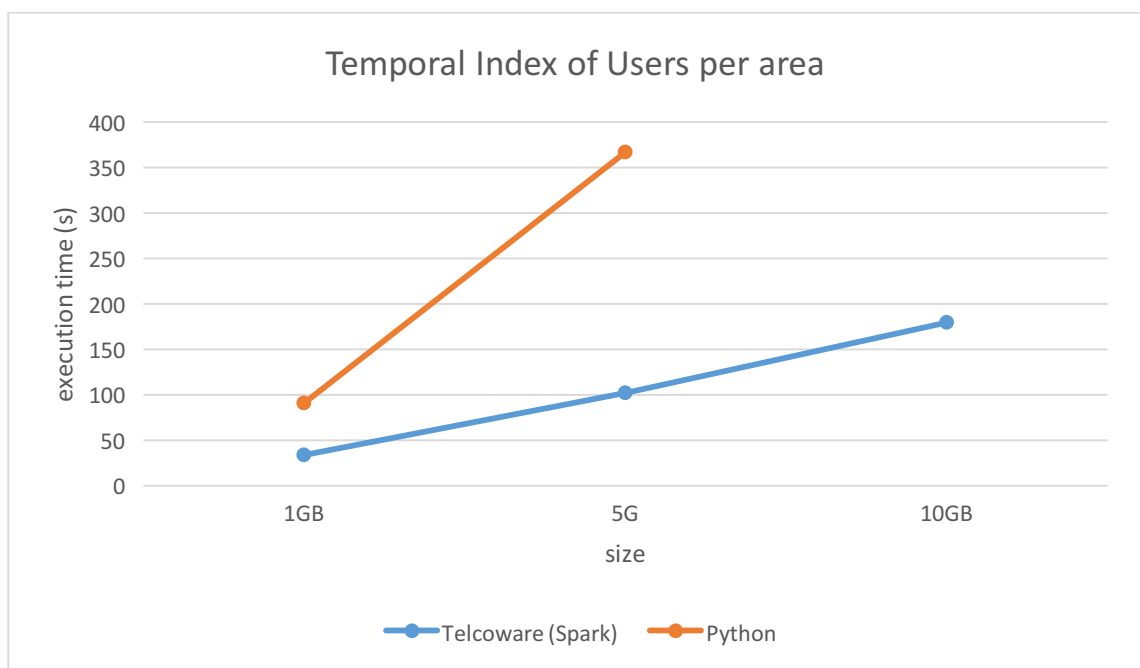
Κτίσιμο Ευρετηρίου Κλήσεων ανά χρήστη



Σχήμα 7.2 Χρόνος εκτέλεσης κτισίματος ευρετηρίου ανάλογα με το μέγεθος των δεδομένων

Παρατηρείται από το Σχήμα 7.2 ότι καθώς το μέγεθος των δεδομένων μεγαλώνει, ο χρόνος εκτέλεσης στο TelcoWare αυξάνεται γραμμικά. Το αντίστοιχο παράδειγμα αξιοποιώντας εντολές UNIX και Python είναι σχετικά εύκολο να επιτευχθεί με χρήση ενδιάμεσων αρχείων. Όπως όμως απεικονίζεται από την γραφική παράσταση δεν είναι αρκετά αποδοτικά και ο χρόνος εκτέλεσης τους αυξάνεται εκθετικά με το μέγεθος των δεδομένων εισόδου.

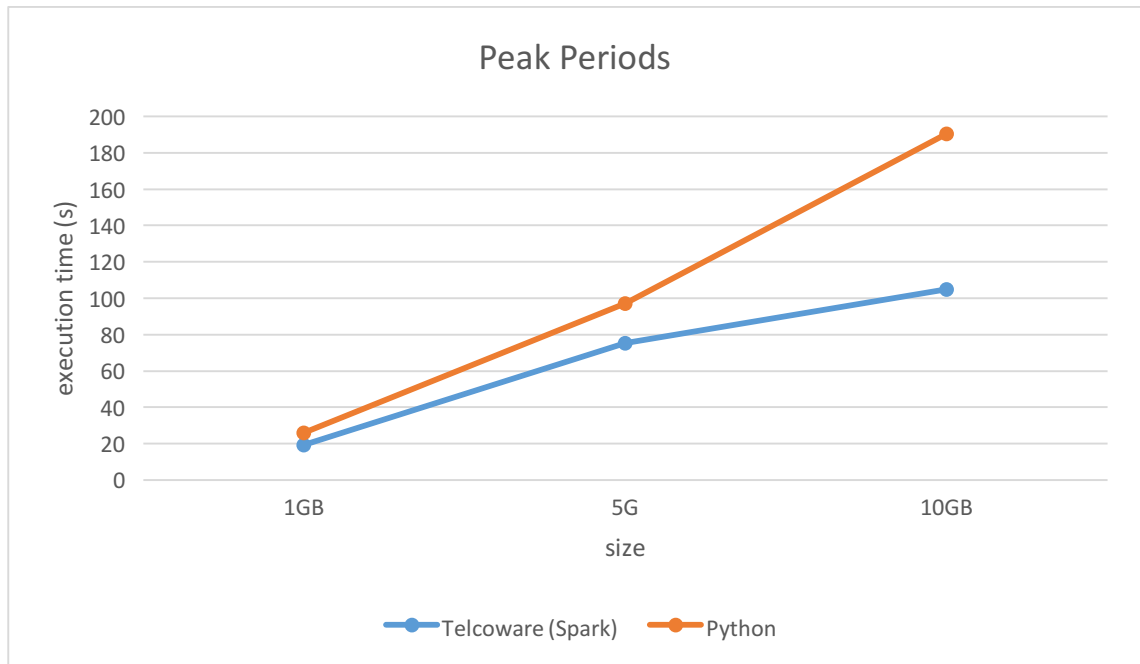
Χρονικό Ευρετήριο Χρηστών ανά Περιοχή



Σχήμα 7.3 Υπολογισμός χρηστών που συχνάζουν στην ίδια περιοχή

Από την εκτέλεση των εργασιών φαίνεται στο Σχήμα 7.3 πώς ο χρόνος εκτέλεσης στη Python αυξάνεται δραματικά απότομα με την αύξηση του μεγέθους του συνόλου δεδομένων. Αντίθετα, με το Spark, όπως θα ανέμενε κανείς, υπάρχει μια πιο ομαλή αύξηση του χρόνου εκτέλεσης.

Χρονικό Ευρετήριο Αριθμού Χρηστών στο Δίκτυο

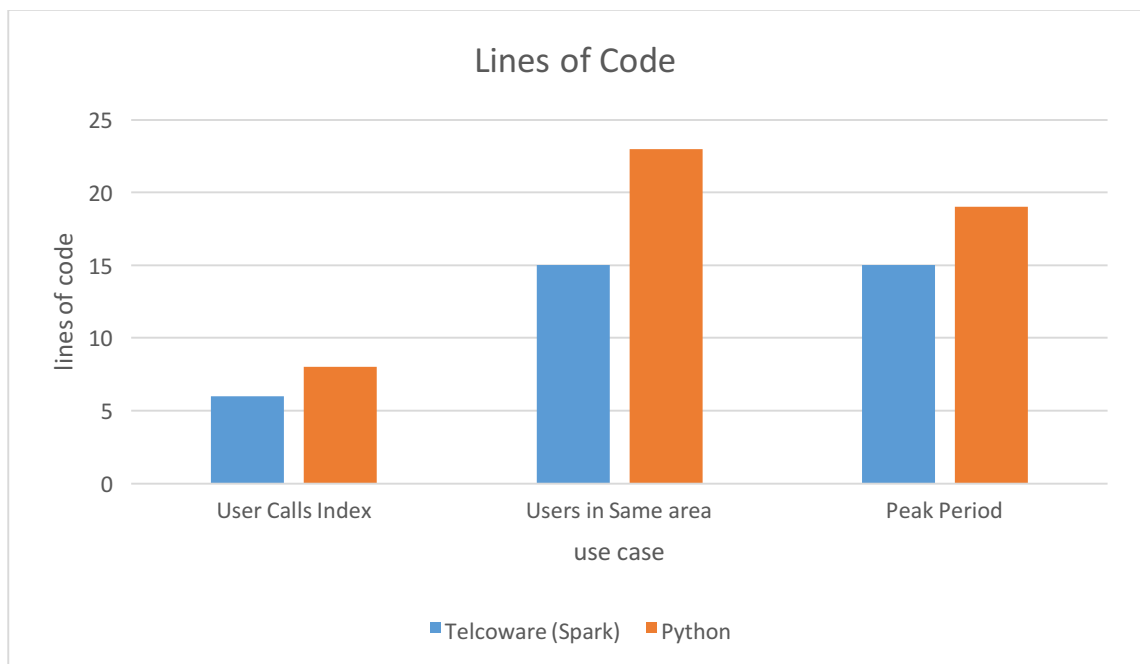


Σχήμα 7.4 Χρόνος εκτέλεσης υπολογισμού ωρών αιχμής στο δίκτυο

Η παραπάνω γραφική (Σχήμα 7.4) παράσταση παρουσιάζει τον χρόνο εκτέλεσης υπολογισμού ωρών αιχμής στο δίκτυο. Όπως φαίνεται, στο σύστημα TelcoWare ο χρόνος εκτέλεσης αυξάνεται λογαριθμικά με το μέγεθος του συνόλου δεδομένων εισόδου. Αντίθετα, το παραδοσιακό εργαλείο Python αυξάνει τον χρόνο εκτέλεσης του εκθετικά.

Η κατανεμημένη φύση του TelcoWare συντείνει στην πιο αποδοτική ανάλυση των δεδομένων εισόδου. Ως αποτέλεσμα ο αλγόριθμος αυτός μπορεί να αξιοποιεί τις ροές δεδομένων του τηλεπικοινωνιακού παροχέα για να εκπαιδεύει γρήγορα και αποτελεσματικά ένα μοντέλο πρόγνωσης ωρών αιχμής του τηλεπικοινωνιακού δικτύου σε σχεδόν πραγματικό χρόνο και να ενημερώνει τους διαχειριστές για τυχόν ανωμαλίες οι οποίες υπολογίζονται με μεγάλη πιθανότητα να προκύψουν.

Γραμμές Κώδικα



Σχήμα 7.5 Σύγκριση γραμμών κώδικα μια αφελής υλοποίησης (naïve implementation) που απαιτούνται για την επίτευξη του επιθυμητού αποτελέσματος

Όπως παρατηρείται από το Σχήμα 7.5 για την υλοποίηση των αλγορίθμων απαιτούνται συνήθως λιγότερες γραμμές κώδικα στο TelcoWare. Αυτό όμως δεν είναι το μόνο όφελος. Αν και με λιγότερες γραμμές, εντούτοις προσφέρει out-of-the-box κατανομημένη επεξεργασία η οποία μειώνει σε μεγάλο βαθμό τον χρόνο εκτέλεσης τους.

Κεφάλαιο 8

Συμπεράσματα και Μελλοντικές Επεκτάσεις

8.1	Συμπεράσματα	61
8.2	Μελλοντικές Επεκτάσεις	63

8.1 Συμπεράσματα

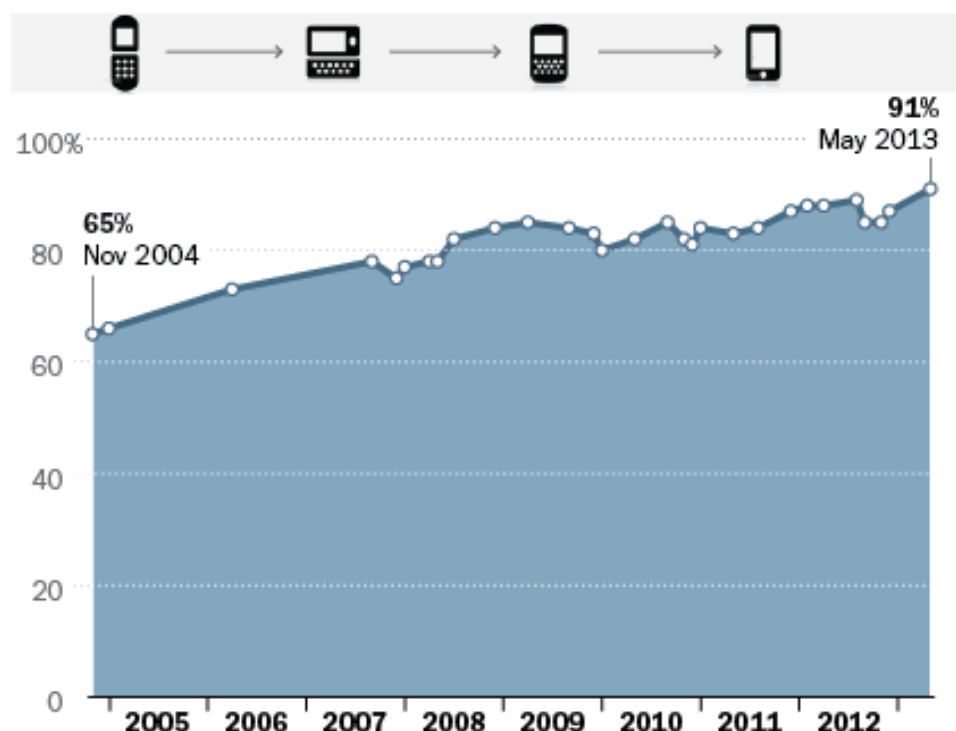
Στα πλαίσια αυτής της διπλωματικής εργασίας παρουσιάστηκε το TelcoWare, μια επεκτάσιμη αρχιτεκτονική για την επεξεργασία μεγάλων τηλεπικοινωνιακών δεδομένων. Σχεδιάστηκε και αναπτύχθηκε ολόκληρη η υποδομή παρέχοντας μια διεπαφή η οποία επιτρέπει την πρόσθεση νέων συστατικών αλλά και πρόσθεση ροών δεδομένων.

Είναι γεγονός ότι φτάσαμε σε μια εποχή όπου ο καθένας μας μπορεί να έχει στη διάθεση του μια κινητή συσκευή, είτε έξυπνη είτε όχι. Σχετικά πρόσφατες έρευνες που έχουν γίνει στην Αμερική, 91% των ενηλίκων έχουν και χρησιμοποιούν κινητό τηλέφωνο. Η αύξηση του ποσοστού αυτού μέσα στα χρόνια φαίνεται στο Σχήμα 8.1. Επίσης είχε αναφερθεί στην εισαγωγή ότι ο αριθμός των κατόχων κινητών τηλεφώνων εκτιμάται να φτάσει τα 5.47 δισεκατομμύρια μέχρι το 2018. Καθημερινώς χρησιμοποιούμε τα κινητά μας τηλέφωνα για να επικοινωνήσουμε με άλλα άτομα. Όλες οι κινήσεις που κάνουμε μέσω των συσκευών έχουν όμως το αντίστοιχο κόστος και εμπλέκουν το δίκτυο του τηλεπικοινωνιακού παροχέα. Οι πελάτες πάντα ψάχνουν να βρουν τον παροχέα που θα τους προσφέρει όσες υπηρεσίες χρειάζονται στην καλύτερη αναλογία ποιότητας και χρέωσης υπηρεσίας.

Μέσω μια πλατφόρμας ανάλυσης τηλεπικοινωνιακών δεδομένων, οι παροχείς μπορούν να αξιοποιήσουν όλη την πληροφορία που ρέει στα συστήματα τους αποκομίζοντας έτσι πολύτιμα στατιστικά σχετικά με τους πελάτες και το δίκτυο. Το TelcoWare έρχεται

Cellphone Ownership, 2004-2013

Percentage of American adults who own a cell phone



Source: Pew Research Center's Internet & American Life Project, April 17-May 19, 2013 Tracking Survey. Interviews were conducted in English and Spanish and on landline and cell phones. Margin of error is +/-2.3 percentage points based on all adults (n=2,252).

PEW RESEARCH CENTER

Σχήμα 8.1 Ποσοστό ενήλικων ατόμων που κατέχουν κινητό τηλέφωνο στην Αμερική σύμφωνα με έρευνες του κέντρου Pew Research Center³⁶

δυναμικά να παρέχει την όλη υποδομή που απαιτείται και μια επεκτάσιμη αρχιτεκτονική στην οποία μπορούν να προστεθούν περισσότερες εργασίες ανάλυσης ανάλογο με τις ανάγκες του κάθε παροχέα ξεχωριστά. Επίσης το TelcoWare είναι επεκτάσιμο και από την άποψη της παρουσίασης των αποτελεσμάτων αφού παρέχει μια διαδικτυακή προγραμματιστική διεπαφή (Web2.0 API) πάνω στην οποία μπορούν να κτιστούν άπειρες εφαρμογές από τρίτους προγραμματιστές. Παράδειγμα μια τέτοιας εφαρμογής

³⁶ Πηγή της έρευνας - <http://www.pewresearch.org/fact-tank/2013/06/06/cell-phone-ownership-hits-91-of-adults/>

είναι το TelcoWare Viewer το οποίο επίσης παρουσιάστηκε στα πλαίσια αυτής της διπλωματικής εργασίας.

Η παρουσίαση των δεδομένων σε ένα γραφικό περιβάλλον συμβάλλει στην καλύτερη κατανόηση των αναγκών των χρηστών αλλά και το δικτύου. Ως αποτέλεσμα συνεισφέρει στην λήψη σημαντικών αποφάσεων που αφορούν τις υπηρεσίες ενός παροχέα και την διαφήμιση τους. Μέσω του συστήματος είναι επίσης εφικτό να εκτελεστούν εργασίες μηχανικής μάθησης με σκοπό την πρόβλεψη της συμπεριφοράς μια ομάδας πελατών ή και ενός συγκεκριμένου πελάτη αλλά και για την πρόβλεψη συμφόρησης στο δίκτυο.

Τέλος, το TelcoWare σε αντίθεση με άλλα συστήματα που ήδη υπάρχουν, αποτελεί μian επεκτάσιμη υλοποίηση η οποία επιτρέπει την εξατομίκευση των αποτελεσμάτων που μπορεί να προσφέρει. Σαν πρώιμο σύστημα, το TelcoWare αποτελεί μια πολύ καλή βάση για την περαιτέρω ανάπτυξη του μέχρι βιομηχανικού επιπέδου αλλά και για την ανάπτυξη περισσότερων εφαρμογών παρουσίασης αποτελεσμάτων προσφέροντας μian έτοιμη out-of-the-box εμπειρία για τους λιγότερο απαιτητικούς.

8.2 Μελλοντικές Επεκτάσεις

Το σύστημα TelcoWare βρίσκεται ακόμα στα αρχικά του στάδια και εξίσου υπάρχουν πολλές επεκτάσεις που θα μπορούσαν να γίνουν. Αν και από πλευράς αρχιτεκτονικής είναι σε καλό σημείο, πρέπει πλέον να αναπτυχθούν εφαρμογές μέσω των οποίων να γίνει ευκολότερη η υποβολή νέων εργασιών ανάλυσης.

Επίσης, σε επόμενο βήμα θα πρέπει μέσω συζήτησης με τηλεπικοινωνιακούς παροχείς να αναδειχθεί κάποιος τελεστής ο οποίος να έχει μεγάλη αξία και αντίκτυπο. Ένας τέτοιος τελεστής θα βοηθούσε στην πιο λεπτομερή ανάλυση και ανάπτυξη του συστήματος εντοπίζοντας σημεία που δεν είναι αρκετά αφαιρετικά.

Στα πλαίσια αυτής της διπλωματικής μελετήθηκαν μόνο τα CDR δεδομένα στα πλαίσια της αποτίμησης. Το σύστημα θα πρέπει να δοκιμαστεί και κάτω από άλλες ροές εισόδου έτσι ώστε να εγγυηθεί η σταθερότητα του και να επιτραπεί η πλήρης λειτουργία του στο τηλεπικοινωνιακό παροχέα.

Τέλος, υπάρχει η ανάγκη σχεδίασης μιας καλύτερης τεχνικής διαχείρισης των αποτελεσμάτων αφού μέχρι στιγμής η δομή με την οποία αποθηκεύονται δεν υποδεικνύει τον τύπο τους. Υπάρχει δηλαδή η ανάγκη για διάκριση μεταξύ αριθμητικών, χωρικών, χρονικών και αθροιστικών αποτελεσμάτων.

Βιβλιογραφία

- [1] Big Data to Improve 'Churn' Analysis in the Telecoms Industry. [Online]. <http://www.reply.eu/en/topics/big-data-and-analytics/big-data-to-improve-churn-analysis-in-the-telecoms-industry>
- [2] Mike Ferguson. Architecting A Big Data Platform for Analytics. [Online]. http://citia.co.uk/content/files/architecting-a-big-data-platform-for-analytics_24606569.pdf
- [3] Sameer Nori. Using big data Hadoop to reduce customer churn. [Online]. <http://www.rcrwireless.com/2014/11/24/opinion/reader-forum-how-to-use-big-data-hadoop-to-reduce-telecom-customer-churn-tag10>
- [4] Borja Balle, Bernardino Casas, Alex Catarineu, Ricard Gavald`a, and David Manzano-Macho, "The Architecture of a Churn Prediction System Based on Stream Mining," Universitat Polit`ecnica de Catalunya,.
- [5] Ari Banerjee, "Big Data & Advanced Analytics: A multi-billion-dollar revenue opportunity," Huawei,.
- [6] Josef Bajada, "Mobile Positioning for Location Dependent Services in GSM Networks," Department of Computer Science and AI, University of Malta, M.Sc Thesis 2003. [Online]. <http://www.cs.um.edu.mt/~csaw/CSAW03/Proceedings/MobilePositioning.pdf>
- [7] Shiming Zhang, Yin Yang, Wei Fan, Liang Lan, and Mingxuan Yuan, "OceanRT: Real-time Analytics over Large Temporal Data," *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, pp. 1099-1102, 2014.
- [8] Eric Bouillet et al., "Processing 6 Billion CDRs/Day: From Research to Production (Experience Report)," *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, pp. 264-267, 2012.
- [9] Spark Homepage. [Online]. <http://spark.apache.org>
- [10] Matei Zaharia et al., "Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing," *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, pp. 2-2, 2012.
- [11] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Commun. ACM*, pp. 107-113, 2008.

- [12] Matei Zaharia et al., "Discretized Streams: Fault-tolerant Streaming Computation at Scale," *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, pp. 423-438, 2013.
- [13] Reynold S. Xin et al., "Shark: SQL and Rich Analytics at Scale," *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pp. 13-24, 2013.
- [14] Sanjay Ghemawat, Shun-Tak Leung, and Howard Gobioff, "The Google File System," *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, pp. 29-43, 2003.
- [15] Call Detail Records (CDR) - Wikipedia. [Online].
http://en.wikipedia.org/wiki/Call_detail_record
- [16] Enterprise Big Data. [Online]. http://wikibon.org/wiki/v/Enterprise_Big-data
- [17] Yan B. Gao. Finding a Goldmine in Cellular Data: A New Opportunity for Telecom in the Big Data Era. [Online]. <http://www.accenture.com/us-en/blogs/technology-blog/archive/2014/05/12/finding-a-goldmine-in-cellular-data-a-new-opportunity-for-telecom-in-the-big-data-era.aspx>
- [18] Zeng Zhi and Yang Yi. Unlocking the Potential of Big Data to Stop Telco Customer Churn. [Online]. http://www.telecomasia.net/pdf/ZTE_071713.pdf
- [19] Telco Big Data Analytics - HP. [Online].
<http://www8.hp.com/us/en/industries/communications-media-entertainment.html?compURI=1089590#.VVNDC9Oqqko>
- [20] Demetrios Zeinalipour Yazti and Shonali Krishnaswamy, "Mobile Big Data Analytics: Research, Practice, and Opportunities," *Proceedings of the 2014 IEEE 15th International Conference on Mobile Data Management - Volume 01*, pp. 1-2, 2014.
- [21] Jonathan Magnusson and Tor Kvernvik, "Subscriber Classification Within Telecom Networks Utilizing Big Data Technologies and Machine Learning," *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, pp. 77-84.